DOCUMENT RESUME

ED 057 824 LI 003 321

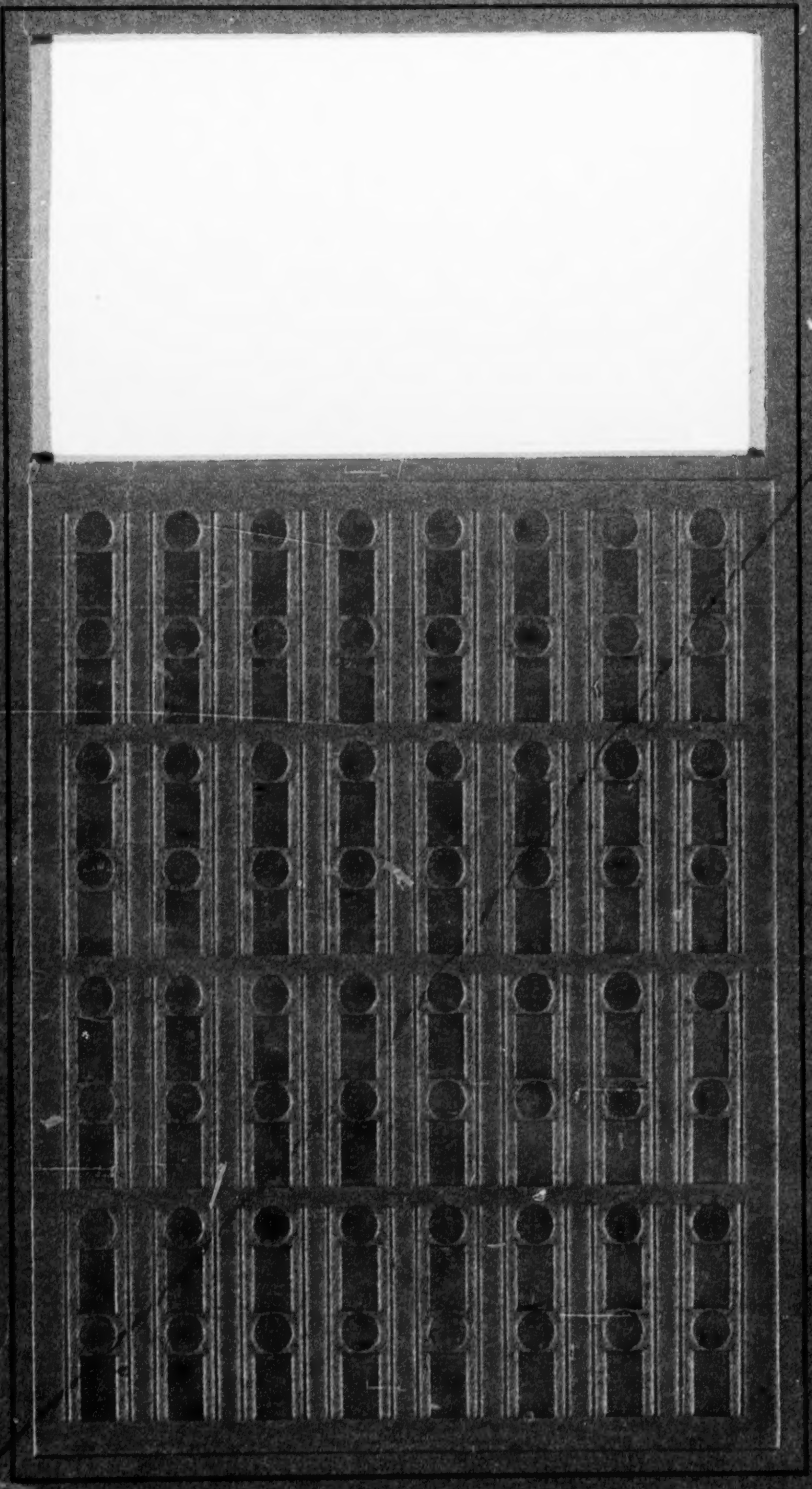
AUTHOR	Kurtz, Peter; And Others
TITLE	On-Line Retrieval II.
INSTITUTION	Informatics, Inc., Bethesda, Md.
SPONS AGENCY	Rome Air Development Center, Griffiss AFB, N.Y.
REPORT NO	TR-71-1202-1
PUB DATE	23 May 71
NOTE	430p.; (5 References)
EDRS PRICE	MF-$0.65 HC-$16.45
DESCRIPTORS	*Automatic Indexing; Classification; *Computer Programs; Indexes (Locaters); Information Processing; *Information Retrieval; *Information Systems; *On Line Systems
IDENTIFIERS	*On Line Retrieval Systems

ABSTRACT

This report is concerned with the implementation of two interrelated computer systems: an automatic document analysis and classification package, and an on-line interactive information retrieval system which utilizes the information gathered during the automatic classification phase. Well-known techniques developed by Salton and Dennis have been adapted for the automatic classification. This work is an extension of an effort begun under contract F30602-69-C-0038, (See ED 045 128). For those interested only in performing retrievals from the on-line console, a short instructive manual is provided in the form of Appendix B of this report. Since this report is to serve as a maintenance manual as well as a description of the work performed and general findings, program listings and other detailed information have been included. (Author)

ED057824

informatics inc.



LI 003 321

1

Wii ____________

Copy ______

Submitted to:
ROME AIR DEVELOPMENT CENTER (INTD)
Air Force Systems Command
Griffiss Air Force Base
New York

PREPARED UNDER CONTRACT
F30602-70-C-0199

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

ON-LINE RETRIEVAL II

ED057824

TR-71-1202-1

Peter Kurtz
Thomas C. Lowe
Thomas A. Bakert

23 May 1971

LI 003 321

informatics inc.
4720 Montgomery Lane
Bethesda, Maryland 20014

FOREWORD

This document is the final Technical Documentary Report for contract F30602-70-C-0199 for the Rome Air Development Center, Griffiss Air Force Base, New York. It documents in detail the implementation of an automatic document classification system and an on-line information retrieval system.

ABSTRACT

This report is concerned with the implementation of two interrelated computer systems: an automatic document analysis and classification package, and an on-line interactive information retrieval system which utilizes the information gathered during the automatic classification phase. Well-known techniques developed by Salton and Dennis have been adapted for the automatic classification. This work is an extension of an effort begun under contract F30602-69-C-0038.

For those interested only in performing retrievals from the on-line console, a short instructive manual is provided in the form of Appendix B of this report. Since this report is to serve as a maintenance manual as well as a description of the work performed and general findings, program listings and other detailed information have been included.

TABLE OF CONTENTS

Section	Title	Page
SECTION I	INTRODUCTION	I- 1
SECTION II	PROJECT OBJECTIVES	II- 1
II. 1	General Objectives	II- 1
II. 2	Specific Objectives: the Batch System	II- 2
II. 3	Specific Objectives: the On-Line System	II- 4
II. 4	Summary of Objectives	II- 8
SECTION 3	SYSTEM DESCRIPTION: THE BATCH SYSTEM	III- 1
III. 1	System Background and Introduction to System Concepts	III- 1
III. 2	Concordance Generation	III- 9
III. 3	The Statistical Filter Package	III-10
III. 4	Creation of the Concept Dictionary (Keyword List)	III-11
III. 5	Automatic Document Classification	III-12
SECTION IV	SYSTEM DESCRIPTION: THE ON-LINE SYSTEM	IV- 1
IV. 1	The Previous On-Line System and the Present One Compared	IV-1
IV. 2	Motivation for Reprogramming the Previous System	IV- 3
IV. 2. 1	Speed of Operation	IV- 3
IV. 2. 2	Core Requirements Problems	IV- 4
IV. 2. 3	Access to Large On-Line Files	IV- 5
IV. 2. 4	Ease of Modularization and Overlay Structures	IV- 5
IV. 2. 5	Language of the Reprogramming Effort	IV- 6
IV. 3	Present System Status	IV- 7
IV. 4	Sample Man-Machine Dialogues with Running Commentary	IV- 8
IV. 4. 1	Man-Machine Dialogue: The "Tyro" User	IV- 8
IV. 4. 2	Man-Machine Dialogue: Experienced Users	IV-11

TABLE OF CONTENTS (Cont'd)

Section	Title	Page
IV.4.2.1	Complete Sophisticated Dialogue	IV-11
IV.4.2.2	Dialogue Portions Illustrating Some of the System Features	IV-16
SECTION V	THE BATCH SYSTEM: IMPLEMENTATION DETAILS AND EXPERIMENTAL RESULTS	V- 1
V.1	Introduction, System Overview, and Flow of Operation Code	V- 1
V.2	Conversion of the Original Source Documents	V- 1
V.2.1	Introduction	V- 1
V.2.2	Flowchart and Program Description: CONVRT	V- 4
V.2.2.1	Program Description	V- 4
V.2.2.2	Definitions of Data Descriptions Used In CONVRT	V-18
V.2.2.3	Preparation for a CONVRT Run	V-22
V.2.2.4	Listing of Program CONVRT	V-25
V.3	Abstract-Bibliography Separation	V-38
V.3.1	Purpose	V-38
V.3.2	Flowchart Comments for GURGLE	V-38
V.3.3	Summary of GURGLE Production	V-44
V.4	Micro-Concordance Generation	V-66
V.4.1	Design Considerations for the Micro-Concordance Generator	V-66
V.4.2	Implementation Details	V-67
V.4.3	Operation of the Micro-Concordance Generator	V-79
V.4.4	Listing of MICCON Items, Tables, File Names and Other Implementation Data	V-79
V.5	The Statistical Filter	V-101
V.5.1	Derivation of the Statistical Filter Algorithm	V-101
V.5.2	Establishing the Content-Stem-Cut-Off Levels	V-104
V.5.2.1	A Property of Dennis' Measure	V-104
V.5.2.2	Establishing Cut-Off: Experimental Observations	V-106
V.5.3	Program Descriptions: MRGSF1 and SF2.B	V-112
V.5.3.1	Phase I of the Filter (MRGSF1)	V-112

TABLE OF CONTENTS (Cont'd)

Section	Title	Page
V.5.3.2	Phase II of the Filter (SF2.B)	V-113
V.5.3.3	Flowcharts for the Statistical Filter Programs, Phase I	V-113
V.5.3.4	Flowcharts for the Statistical Filter Programs, Phase II	V-116
V.5.4	Run Times and Process Charts for the Statistical Filter	V-128
V.6	Concept Dictionary Generation (Stem Clustering)	V-137
V.6.1	Purposes	V-137
V.6.2	Method	V-138
V.6.2.1	Flowchart Comments: CLUSTR.0	V-139
V.6.2.2	Production Run Summary: CLUSTR.0	V-144
V.6.2.3	Operational Procedure: CLUSTR.0	V-144
V.6.2.4	Flowchart Comments: CLUSTR.1	V-144
V.6.2.5	Operational Procedure: CLUSTR.1	V-150
V.6.2.6	Production Run Summary: CLUSTR.1	V-151
V.6.2.7	Flowchart Comments: CLUSTR.2	V-151
V.6.2.8	Operational Procedure: CLUSTR.2	V-152
V.6.2.9	Production Run Summary: CLUSTR.2	V-152
V.6.3	Summary of Concept-Dictionary Generation	V-174
V.7	Document Concept Vector Generation	V-176
V.7.1	Techniques	V-176
V.7.2	DCVGEN: Program Description	V-177
V.7.3	Flowchart Comments: DCVGEN	V-177
V.7.4	Operational Procedure: DCVGEN	V-178
V.7.5	Production Summary: DCVGEN	V-186
V.7.6	Introduction to BLD.STR, Inverted-List Builder	V-186
V.7	Flowchart Comments - Routine BLD.STR	V-187
V.8	Summary of the Batch System	V-192

TABLE OF CONTENTS (Cont'd)

Section	Title	Page
SECTION VI	THE ON-LINE SYSTEM: IMPLEMENTATION DETAILS	VI- 1
VI. 1	Introduction to the Implementation of the On-Line System: General Design Philosophy	VI- 2
VI. 2	Functional Description of the On-Line System	VI- 5
VI. 2. 1	The Executive Monitor	VI- 5
VI. 2. 1. 1	Supervisory Routines	VI- 5
VI. 2. 1. 2	Executive Utility Routines	VI-11
VI. 2. 2	User Functions	VI-17
VI. 2. 3	Overlaid Portions of the On-Line System	VI-45
VI. 2. 3. 1	Design Philosophy	VI-45
VI. 2. 3. 2	ISOLate	VI-46
VI. 2. 3. 3	ANALyze	VI-47
VI. 2. 3. 4	LOOKup	VI-48
VI. 2. 3. 5	RETRIV (Document Retrieval)	VI-58
VI. 2. 3. 6	NUMBER	VI-61
VI. 2. 3. 7	CWPAIR	VI-74
VI. 2. 3. 8	OFFline	VI-75
VI. 3	The Structure of the On-Line System	VI-75
VI. 3. 1	General Structure of the System	VI-75
VI. 3. 2	Problems with Global Procedures	VI-76
VI. 3. 3	Saving the Independently-Compiled Routines	VI-77
VI. 3. 4	The On-Line System at GELOAD Time	VI-77
VI. 3. 5	Saving the Links Built During the GELOAD Activity and Subsequently Reloading Them	VI-78
VI. 3. 6	Illustrations of the Overlay Process	VI-79
VI. 4	External Files Used by the Document Retrieval System	VI-79
VI. 4. 1	STEMDICT	VI-79
VI. 4. 2	CONTENTS	VI-87

TABLE OF CONTENTS (Cont'd)

Section	Title	Page
VI.4.3	BCDSMS; BCDLMS	VI-87
VI.5	Detailed Description of the Dialogue Processor Executive	VI-88
VI.5.1	Table Descriptions	VI-88
VI.5.2	Global Variables Description	VI-91
VI.6	Future Direction of the On-Line System	VI-92
VI.6.1	Further Program Segmentation	VI-92
VI.6.2	How to Add Additional Functions	VI-93
VI.6.3	Increasing the Users' Control Over the System	VI-94
SECTION VII	UPDATING THE SYSTEM: RECOMMENDATIONS AND SUGGESTIONS	VII- 1
VII.1	General Considerations	VII- 1
VII.2	Details of Document Form for Input to MIC.CON	VII- 1
VII.3	Detailed Procedure for Limited Updating	VII- 2
VII.4	Possible Problems with Limited Updating	VII- 2
VII.5	Large-Scale Updating of the System	VII- 3
SECTION VIII	PROJECT SUMMARY	VIII- 1
SECTION IX.	REFERENCES	IX- 1
APPENDIX A	INSTALLATION OF THE DOCUMENT RETRIEVAL SYSTEM	A- 1
A.1	The Batch System	A- 1
A.1.1	Machine Configuration Needed	A- 1
A.2	The On-Line System	A- 1
A.2.1	Machine Configuration Needed	A- 1
A.3	System Software	A- 1
A.3.1	Operating System Requirements	A- 1

TABLE OF CONTENTS (Cont'd)

Section	Title	Page
APPENDIX B	USER'S MANUAL: RADC ON-LINE RETRIEVAL SYSTEM	B- 1
B.1	The On-Line Retrieval System	B- 1
B.1.1	Automatic Document Characterization	B- 1
B.1.2	The Interactive Environment	B- 1
B.1.3	Notational Conventions	B- 2
B.2	What the System Will and Will Not Do	B- 3
B.3	How to Use the System	B- 4
B.3.1	How to Sign Onto the System	B- 4
B.3.2	Description of Available Options	B- 5
B.3.2.1	ENDF	B- 5
B.3.2.2	MOD	B- 5
B.3.2.3	DOC	B- 6
B.3.2.4	HELP	B- 6
B.3.2.5	OFF	B- 6
B.3.2.6	CHG	B- 6
B.3.2.7	CON	B- 6
B.3.2.8	RET	B- 6
B.3.2.9	DEL	B- 6
B.3.2.10	SEE	B- 6
B.3.2.11	CLR	B- 6
B.3.2.12	WRD	B- 6
B.3.2.13	DDC	B- 6
B.3.2.14	WGT	B- 6
B.3.2.15	ENDRUN	B- 6
B.3.2.16	BLD	B- 8
B.3.2.17	CHAT	B- 8
B.3.2.18	SKIM	B- 8
B.4	Sample Complete Dialogues with the On-Line System	B- 8

LISTING OF FIGURES

Figure	Title	Page
II. 1	Project process chart	II- 9
III. 1	Three disjoint concept centers	III- 3
.2	Intersecting concepts	III- 4
.3	Two vectors in a 2-dimensional space	III- 5
.4	Geometric computation of Euclidean distance between two vectors in a 2-dimensional space	III- 7
.5	Illustration of error introduced by the use of the Euclidean distance as a correlation measure between 2 sealarly-related vectors	III- 8
V. 1	Overall data flow through the batch system	V- 2
.2	Structure of IBM 360, 9-Truck Tapes	V- 5
.3	Flowchart CONVRT	V- 6
.4	CONVRT production deck set-up	V-14
.5	CONVRT listing	V-26
.6	GURGLE flowchart	V-45
.7	GURGLE listing	V-58
.8	Symbolic layout of PEEKTB	V-69
.9	PRERUN flowchart	V-70
.10	SMEAR flowchart	V-72
.11	PLETAB flowchart	V-74
.12	Punched format of common words and suffixes	V-75
.13	SUFTAB flowchart	V-76
.14	WORD flowchart	V-80
.15	MIC. CON deck set-up	V-86
.15a	MIC. CON listing	V-87
.16	Suggested V_c versus $\bar{g}_c^2 / S_c$	V-107
.17	Stem number by rank of V_c plot	V-108
.18	σ_c versus $\bar{g}_c^2 / S_c^2$ for selected points	V-110
Table V. 1	Rank, stem, V_c, σ_c, $\bar{g}_c^2 / S_c^2$ list for selected points	V-111
.19	MRGSF1 flowchart	V-117
.19a	MRGSF1 listing	V-122

LISTING OF FIGURES (Cont'd)

Figure	Title	Page
V .20	SF2. B flowchart	V-129
.20a	SF2. B listing	V-131
.21	Statistical filter process chart	V-135
.22	Flowchart CLUSTR. 0	V-140
.23	Flowchart CLUSTR. 1	V-147
.24	Flowchart CLUSTR. 2	V-153
.25	Listing CLUSTR. 0	V-155
.26	Listing CLUSTR. 1	V-160
.27	Listing CLUSTR. 2	V-167
.28	First page of the stem dictionary listing	V-175
.29	DCVGEN flowchart	V-179
.29a	Listing DCVGEN	V-181
.30	Flowchart BLD. STR	V-189
Table V. 2	Batch system tape table	V-194
VI. 1	Flowchart DIALOG	VI- 6
.2	Flowchart CHAT	VI- 8
.3	Flowchart OPT	VI-10
.4	Message pointer tables	VI-13
.5	Flowchart YESNO	VI-15
.6	CHG flowchart	VI-20
.7	DDC flowchart	VI-22
.8	DOC flowchart	VI-23
.9	RET flowchart	VI-26
.10	WRD and SEE Flowchart	VI-27
.11	CLR flowchart	VI-28
.12	DEL flowchart	VI-30
.13	MOD flowchart	VI-31
.14	CON flowchart	VI-35
.15	ENDF flowchart	VI-36
.16	BLD flowchart	VI-38
.17	WGT flowchart	VI-39
.18	JOBEND flowchart	VI-40

LISTING OF FIGURES (Cont'd)

Figure	Title	Page
VI. 19	SKIM flowchart	VI-41
.20	SNDABS; SNDBIB flowchart	VI-42
.21	COMTMP flowchart	VI-43
.22	QCVBLD flowchart	VI-44
.23	LOOK flowchart	VI-52
.24	LOOK listing	VI-54
.25	RETRIV flowchart	VI-62
.25a	RETRIV listing	VI-67
.26	Run H*	VI-80
.27	Load map	VI-81
.28	DIALOG processor listing	VI-95
.29	ANALyze listing	VI-128
.30	NUMBER listing	VI-137

SECTION 1

SECTION 1

INTRODUCTION

This document is the final report on the continuing development effort devoted to producing an on-line information retrieval system for the Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, New York. Under this contract two distinct but related program packages have been built: the automatic document characterization system, called the Batch System and a revised version of the man-machine dialogue processor, called the On-Line System. Significant experimentation has been performed within the Batch System. In addition certain experiments have been run which have never been attempted on a data base as large as the one used for this project, 36,000 abstracts of the American Psychological Association. The On-Line System has been entirely reprogrammed to interface with the data generated by the Batch System. In addition, portions of the system which were only demonstrated in the processor built under contract F30602-69-C-0038 have been implemented in full in this version.

In this document, an overview of the Batch System is first presented and certain significant results are described. Next, the revised On-Line System is discussed, mainly in the light of how it differs in significant detail from the previous processor. After these two sections, the Batch System is discussed at some length, including implementation details. Then a more detailed presentation of the On-Line System is made, including suggestions for improvements in machine utilization and for new features which could be added without major revisions. A review of the project and suggested areas for further investigation conclude the report. One of the appendices to this report is a User's Manual: R.A.D.C. On-Line Document Retrieval System, which is provided for the convenience of any who may need to perform retrievals. Computer listings are included, as well as specific instructions for system installation.

SECTION II

PROJECT OBJECTIVES

This section summarizes the objectives of project On-Line Retrieval II. Included are a gross project process chart, sub-sections on both the batch system and the on-line system, and a brief summary of some of the experimental results obtained from each.

II. 1 GENERAL OBJECTIVES

In keeping with the goals of previous research in this area along the same lines, this project had as its principle desired end-product an information storage and retrieval system with the following characteristics:

1) documents to be entered into the system had to be in machine-readable form, but not necessarily in a compatible code structure.
2) no manual intervention was allowed in the analysis of the source data, except for small lists of stop-words and grammatical endings (suffixes).
3) retrieval was to be performed in an interactive man-machine environment, with a system which could adapt to the level of sophistication of the user.
4) the entire system, batch and on-line together, could be reconfigured for a larger or smaller data base, within practical limits, as well as an entirely different document collection, with a very minimum of program changes.

All the goals listed above have been either met or exceeded in the implementation of the retrieval system.

The text of the Psychological Abstracts was available in machine-readable form, but not in an HISI-635 compatible character code. However, the tape formats were known as well as the text delimiters, creating very few real problems in conversion to the tape form necessary for processing on the HISI 635. Some changes were necessary in the packing factors used on the source data tapes: the HISI 635 has an upper bound on tape physical modules of the system may be modified without the need for any modifications in the executive portion.

II. 2 SPECIFIC OBJECTIVES: BATCH SYSTEM

The batch system itself has two related objectives:

1) experiements in automatic document classification
2) support for the on-line system in two areas
    a) data generation
    b) loader programs for on-line files

These topics are discussed further in the paragraphs below.

As far as experimentation goes, the batch system is a stand-alone research tool. By changing one of the programs in the chain of units leading up to the document-concept-vector generator, a researcher may observe the effects of a particular theoretical approach. By and large such experimentation has not gone on in this project to a great degree. The exceptions to this statement involve the "tuning" of the statistical filter programs, and the modifications which had to be made to the stem clustering algorithms in order to run the programs within a reasonable amount of computer time. Details on these topics are available in sections V. 5 and V. 6 of this report.

That it is indeed possible to create an entire system which processes raw documents and results in useful information being generated about these data has been adequately demonstrated. Because of the large size of the information structures generated for each document in the collection, it is not practical to list the automatic classification for each abstract.

For the purpose of this project the most important function of the batch system is the role it plays in supporting the on-line processor.

records when utilizing standard system format records which is smaller than the allowable physical record size on the IBM S/360 tapes.

As no manual intervention was allowed in the document analysis, the batch system (that portion of the system devoted to automatic document classification) received as input the raw documents and returned all the information necessary to load the on-line system. Programs in the batch system performed the following functions:

1) Micro-concordance generation*
2) Master concordance generation and statistical filtration
3) Final statistical filtration and dimension reduction
4) Concept-Dictionary generation (stem clustering)
5) Document-concept-vector generation (automatic document classification)

In addition there are a series of "loader" programs designed to service the on-line system with raw data for retrieval. Each loader program prepares one of the files needed to run the interactive system.

For the interactive retrieval portion of the project an entirely new on-line processor has been built, in order to interface with the file structures built by the batch system. New features and options have also been added to this Dialogue Processor. Changes in programming language, operating environment, and the addition of overlaid code have been combined to yield a more efficient product in terms of machine utilization and response time. Also, the adaptability of the system to the level of sophistication of the user has been enhanced to ease the transition between the beginner and the experienced researcher.

The on-line processor has been designed in a much more modular fashion than its predecessor system, with the result that much experimentation and change may be affected without altering its basic layout. Whole

---

* Micro-concordances are stem-token counts for individual documents.

The batch programs analyze the input documents, generating data structures used during the on-line retrieval. In effect, the combination of the two systems provides a virtually automatic on-line system, capable of being used on almost any data base, provided that the limitations of the data structures are not exceeded. Given the raw documents as input, the programs "produce" the complete system as output.

The second function of the batch programs is direct support for the on-line system, in terms of actual preparation of the data files used during the retrieval process. It is in these programs that the ultimate size of the retrieval base for any particular document collection is determined. By changing compiler or assembler options in the source code, any subset of a collection may be installed in a particular environment. Increasing the retrieval base is accomplished by the same method, up to the limit of the available documents. Of course there exist practical limitation, owing to the available computing resources, such as high-speed memory, disk space, number of tape drives (used during the numerous merges, sort-merges, et cetera). However, these above limitations do not affect the loading of the on-line system as such: only the amount of disk available for on-line files poses any problem.

II. 3 SPECIFIC OBJECTIVES: THE ON-LINE SYSTEM

In order to be deemed satisfactory the on-line system had to meet several objectives:

1) it had to interface easily with the batch system
2) core-memory space had to be kept to a minimum
3) dialogue response time had to be minimized
4) retrieval time had to be minimized
5) system changes should require only minor changes to the executive of the processor
6) additions to the dialogue processor could be made without affecting other operational portions
7) expansion of the data base should not require any more than re-compilation of a few routines

8) the completed on-line system should be available to the user in a "system-loadable" form, i.e., in H* format for the HISI-635

Each of the above topics are treated in more detail in the paragraphs below.

Very few problems were encountered in creating a Dialogue Processor which would interface with the batch portion of the system. Since all of the file structures were known in advance, the on-line processor was simply programmed to accept these files. Some more careful design had to go into the actual file structures for the on-line files, but by and large this objective was met without difficulty.

In a large on-line system core-memory utilization is a great problem. In general, the more information which can be kept in core at any one time the faster the system runs. Swapping information in and out of random-access storage is bound to result in considerable overhead, especially in a multi-programming environment such as the HISI 635. Therefore, the design of the on-line system had to take into account many different factors affecting memory utilization.

The bulk of the Dialogue Processor lies in its internal tables, plus all the executable code for processing the users' requests. The combined length of all the tables and required code is well over 50K of high-speed core memory, clearly an unacceptably large amount. The easiest solution to this problem was to use program links (overlays) for the less-frequently-used sections of code, retaining as core resident those portions which underwent constant use. The system as it now stands uses 29K of core, almost 7K of which is used exclusively by the retrieval programs (without the retrieval portions of the system, 22K will hold the entire system). The overlays are completely transparent to the on-line user.

Dialogue response time is mostly a function of issuing user messages and interacting with the terminal. In addition, the actual instantaneous machine environment is of some concern: batch programs (such as the Dialogue Processor) obtain more CPU cycles per unit of elapsed time than do time-sharing programs, since the former compete with far fewer other users than the latter. Another matter is the question of the actual messages to be sent. If these are available in high-speed memory (core), the time used to issue them is I/O bound by the low-speed terminal. device. For this reason, all messages to be sent to the user are pre-stored in core tables.

Response time for a retrieval is another matter altogether. Response to a given query in terms of elapsed time is dependent upon several factors. In the first place, the instantaneous state of the computer at retrieval processing time (how many users, their individual requirements, etc.), is the over-all governing consideration, as this load factor determines the number of CPU cycles available to the retrieval programs. Another factor is the number of documents against which the query is run.

The matter of the number of documents does not concern fetching the actual documents: the retrieval method precludes this. Instead, what varies as the quantity of documents increases is the length of each inverted list for each of the 1500 concepts. When the information in one list exceeds 64 machine words, more than one disk access is required to fetch the list, thereby increasing the I/O overhead. Also, the more document numbers there are on each inverted list the more disk accesses are required to fetch the relevant document-concept-vectors for correlation. The longer the query typed by the user, the more the probability of a greater number of content-stems being present in the query, increasing dictionary look-up time. Lastly, the access time of the device on which the files reside (especially the document concept vectors) plays a large role in determining the retrieval processing time.

The Dialogue Processor minimizes retrieval time by the use of random-access organization of the inverted lists and the document-concept-vectors, as well as all other files used furing the retrieval phase (the stem-dictionary, bibliographies, abstracts). Considering the trade-offs available between retrieval time and core space used, the on-line system meets present needs very nicely and can easily handle even more data.

Another matter for consideration is the question of on-line access to the terminal. In order to minimize coding changes with different releases of the operating system (GECOS-III), telecommunications access is handled through conversational GEFRC (GEneral File and Record Control). In this way the on-line system uses manufacturer-supported software, avoiding possible re-coding with new releases of the system. Additions and/or expansions to the on-line software are considered next.

One fortunate side-effect of the overlay process chosen to minimize core requirements is that it in effect highly modularizes the Dialogue Processor code, allowing additions and changes to the on-line system without affecting "debugged" and operational portions. Also, an extremely large expansion of the system can be affected, as long as each overlay does not exceed by too great an amount the length of the longest existing link. If a new link exceeds the longest used link, an adjustment must be made in the core limits required to run the job. The next objective to be considered is the question of expansion of the data base itself.

Expansion of the data base beyond the initial group of documents is discussed in some detail in section VII of this report. The on-line system shares one of the features of the batch system in this particular area: changes to the code in the sensitive places are accomplished through compilation of assembly options in the source code. Re-compilation or re-assembly of the specific programs dealing with the number of available documents is the simplest method of modifying the system to run with a larger group of documents. In this respect the on-line system is as adaptable to change as the batch system.

The last objective of the on-line system is that it be available in a form as close to core-image as feasible within the constraints of the operating system (GECOS-III). Again, the use of overlays has led to the use of the "system-loadable" form for program storage, programs being resident on one of the disk packs used to contain the file data for the on-line system. Re-building the system can be accomplished either from the object decks generated during compilation of assembly or directly through a compile and assemble and load series of activities. Details on this process are available in Appendix A to this report (Installation of the Document Retrieval System).

II. 4 SUMMARY OF OBJECTIVES

In the implementation of the batch and on-line systems all the goals enumerated above have been met or exceeded: a complete information storage and retrieval system may be built with the minimum of human intervention; the system can be easily reconfigured to adapt to a different portion of the available data or with entirely new data; few if any program changes are necessary for the system adaptation; the on-line system is flexible, speedy, and provides several levels of approach by users; on-line files are loaded only once and thereafter reside on mass-storage devices; a system-loadable version of the on-line processor is used instead of object decks, in order to speed up getting the system into operation. Figure II. 1 illustrates the gross project process chart, indicating the major steps taken in the creation of the operational retrieval system.

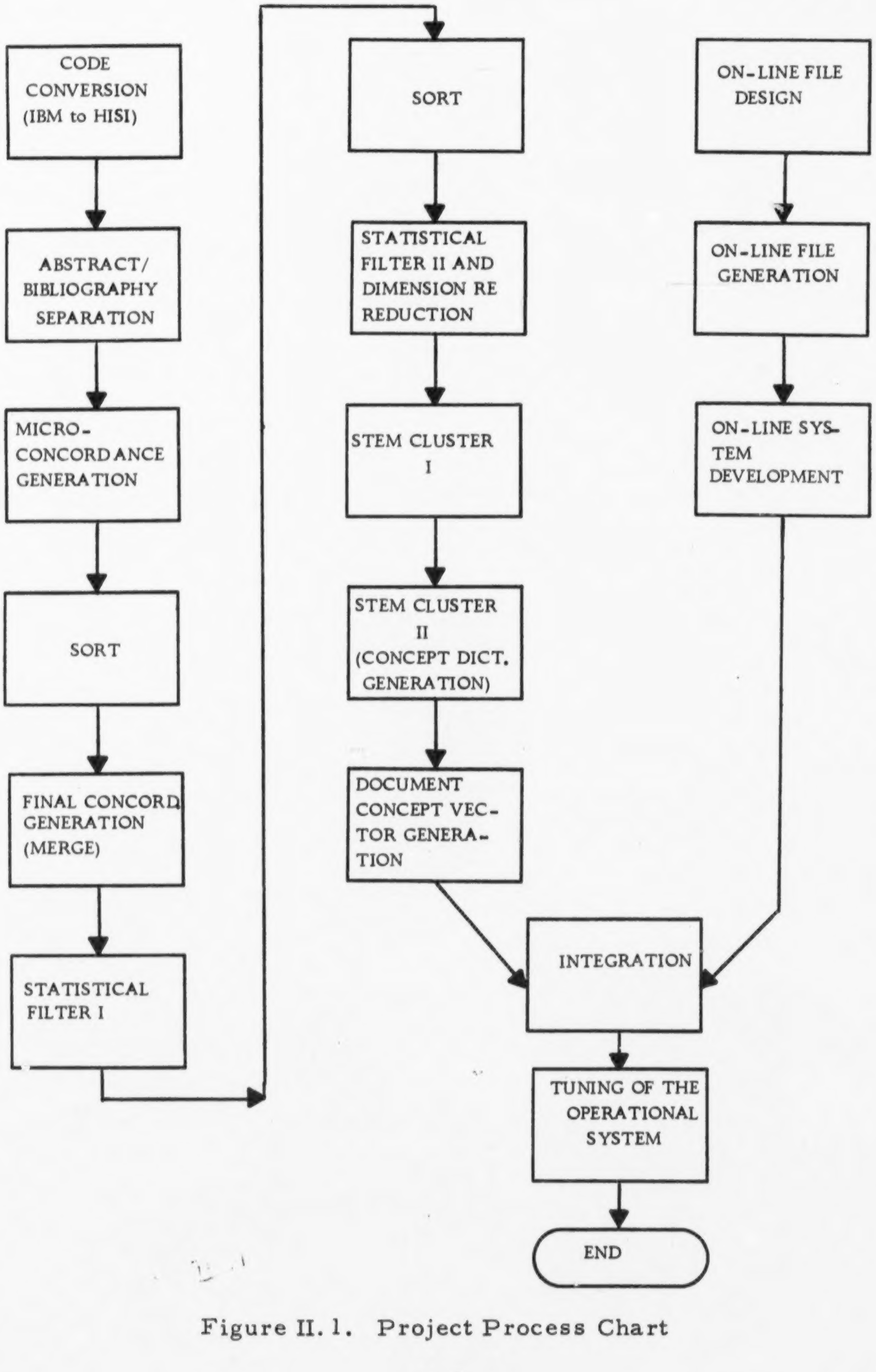

CODE CONVERSION (IBM to HISI)
ABSTRACT/ BIBLIOGRAPHY SEPARATION
MICRO-CONCORDANCE GENERATION
SORT
FINAL CONCORD GENERATION (MERGE)
STATISTICAL FILTER I
SORT
STATISTICAL FILTER II AND DIMENSION RE REDUCTION
STEM CLUSTER I
STEM CLUSTER II (CONCEPT DICT. GENERATION)
DOCUMENT CONCEPT VECTOR GENERATION
ON-LINE FILE DESIGN
ON-LINE FILE GENERATION
ON-LINE SYSTEM DEVELOPMENT
INTEGRATION
TUNING OF THE OPERATIONAL SYSTEM
END

Figure II. 1. Project Process Chart

SECTION III

SYSTEM DESCRIPTION: THE BATCH SYSTEM

III. 1 SYSTEM BACKGROUND AND INTRODUCTION TO SYSTEM CONCEPTS

The stated aim of the batch system is to provide a method whereby a document collection can be automatically classified with respect to its contents. The information gathered during the batch phase is then used in building an on-line retrieval system. However, there is another factor to be considered with respect to the Batch System: experimentation. For this reason, much of the Batch System is designed to be easily modified during the course of operational production, so that various experiments in automatic document classification may be made. On the other hand the actual production runs using the Batch System have all been made in support of the On-Line System.

The technical approach of this system is best termed the "concept vector" method.(1) Adequate justification for this approach has already made in the literature and in previous Technical Reports, (RADC-TR-69-304, RADC-TR-70-44) so that this section will merely serve as a review of the technology involved. First there will be a brief discussion of the mathematical model used to represent the document collection and queries made against it, then in turn discussions about concordance generation, statistical filtration, and automatic keyword generation (stem clustering).

Documents and queries alike are characterized by a 1501-dimension hyperspace, each dimension of which represents one of the concept centers in the collection. Another way of looking at the model is to interpret each of the dimensions as a semantic domain. In theory the 1501 sets of semantic domains should be disjoint, but the "tightness" of a particular document collection affects the amount of potential overlap between the sets. Also the quality of the clustering algorithm used to generate the sets of concept centers affects to a great extent the volume of the hyperspace included in the union of two or more sets. Figure III.1 represents three disjoint concept centers, the ideal situation, while figure III.2 illustrates what actually occurs in practice.

Now, if each document in the collection is characterized a weighted mapping of its content stems into the concept space, there is a point in the hyperspace which uniquely classifies a particular document. If, in turn, some measure is used to compute the "distance" between documents, the value of the measure becomes a determination of the semantic similarity between documents. The same evaluation measures hold true for the similarity between queries and documents.

If, as mentioned above, each concept is assigned one dimension of an N-dimensional hyperspace, and if each document lies at a fixed point within this space, vector analysis may be used to determine whether a query matches documents.

For the sake of simplicity and ease of representation the following discussion and examples will be limited to 2-dimensional space. Figure III.3 illustrates two vectors, $\overline{X} = [5, 8]$ and $\overline{Y} = [8, 7]$. If $\overline{X}$ represents a document and $\overline{Y}$ a query, then, as mentioned above, some measure of correlation must be found which determines the closeness of the match.

For simplicity the Euclidean distance between the vectors is most intuitively attractive. In a space of N dimensions the Euclidean distance from the tip of one vector $(\overline{X})$ to another $(\overline{Y})$ is

$$ED = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

if both vectors share a common origin.

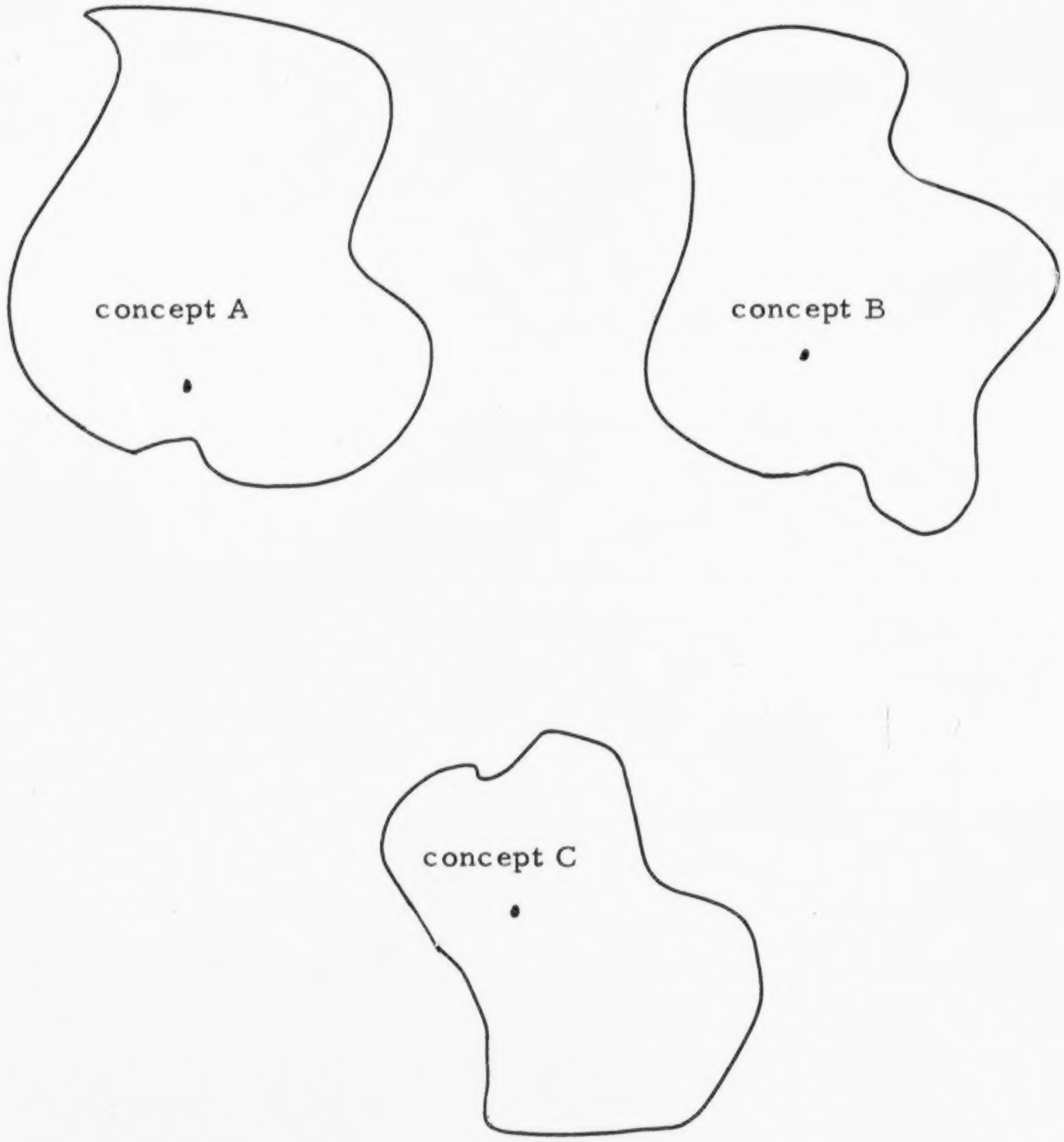

concept A
concept B
concept C

Figure III. 1. Disjoint Concepts

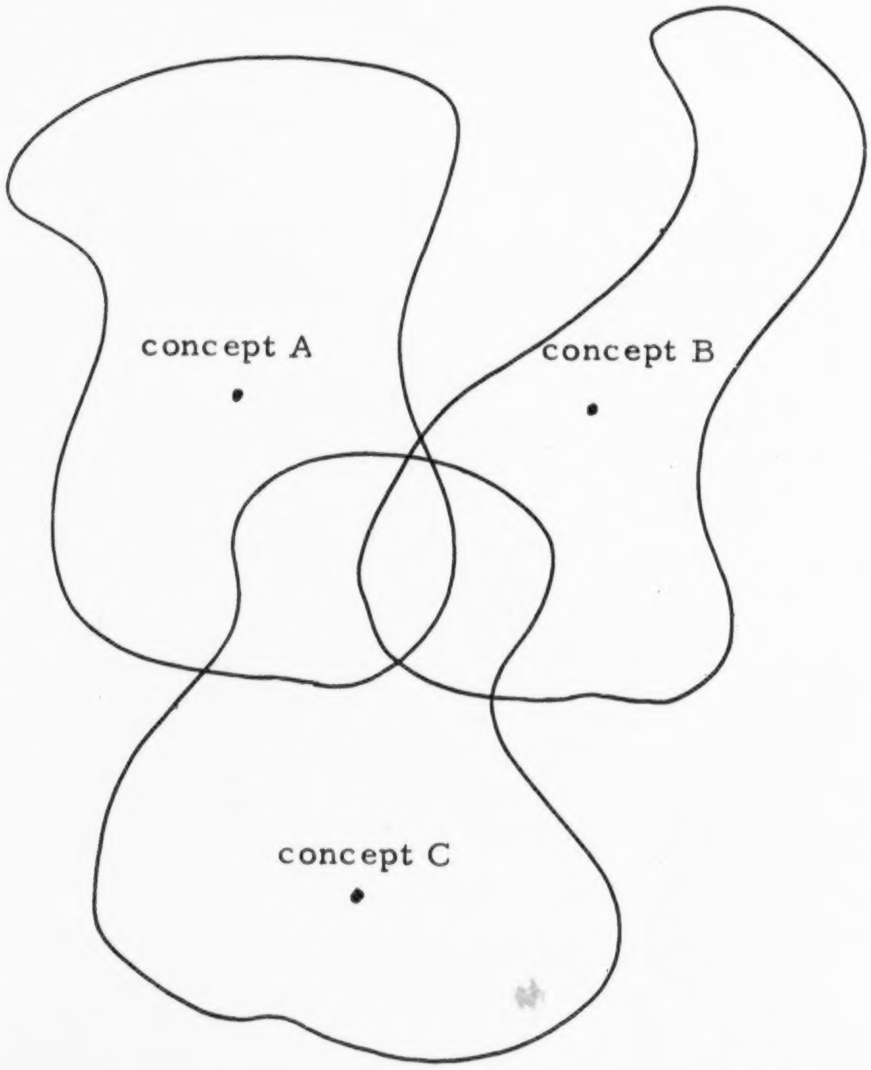

concept A
concept B
concept C

Figure III. 2. Intersecting Concepts

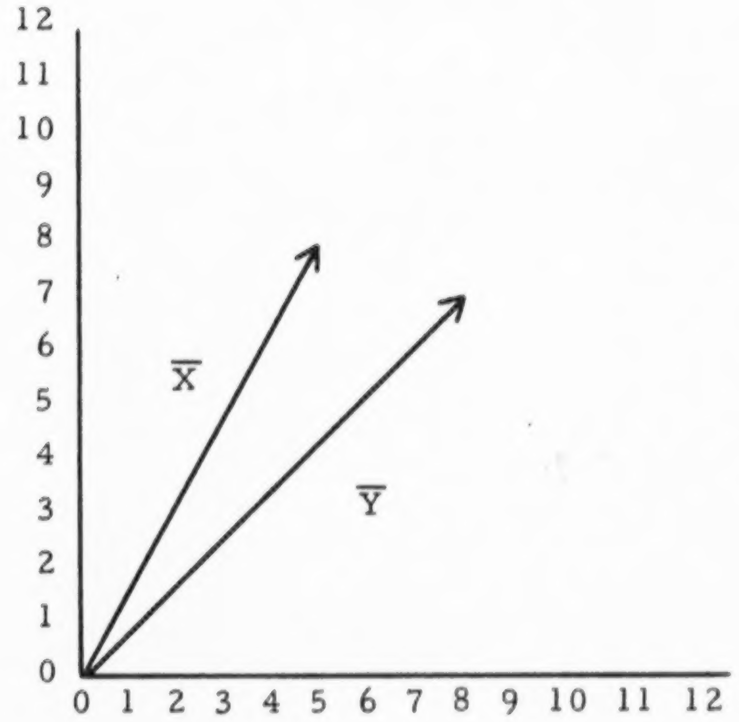


Figure III. 3. Two Vectors In A Concept Space Of Two Dimensions

Applying this formula to the values in figure III. 3, a distance of 3.16 is obtained (figure III. 4). As a correlation measure, this method has several disadvantages:

1. The range of distances is not normalized. This could be corrected once the maximum distance (D) is known, but D would have to be computed between all documents first.
2. More importantly, vectors with equal ratios of concepts, but differing in magnitudes should be considered the "same." Figure III. 5 illustrates the error caused by scalarly-related vectors. Because of this error, Euclidean distance cannot be used as a correlation measure.

On the other hand, the cosine of the hyperangle between vectors meets both of the objections to Euclidean distance mentioned above:

1. Cosine lies between zero and unity (in the range $0^\circ \leq \theta \leq 90^\circ$), providing normalized values.
2. Since the cosine of the hyperangle between vectors is a function of the ration of the vector components, scalarly-related vectors such as the ones in Figure III. 5 have a cosine of 1.000, representing the highest possible correlation.

Another valid correlation measure also exists (Tanimoto)(4) which performs vector multiplication to obtain a similarity measure much like the cosine correlation. The two computation methods are illustrated in the formulas below:

1. Cosine, where θ = the hyperangle in n-space and $\overline{D}_1 = [d^1_1, d^2_1, \ldots, d^N_1]$, $\overline{D}_2 = [d^1_2, d^2_2, \ldots, d^N_2]$

   formula (2)

$$\text{Cos } \theta = \frac{\sum_{i=1}^{N} (d^i_1 \cdot d^i_2)}{\sqrt{\sum_{i=1}^{N} (d^i_1)^2 \cdot \sum_{i=1}^{N} (d^i_2)^2}}$$

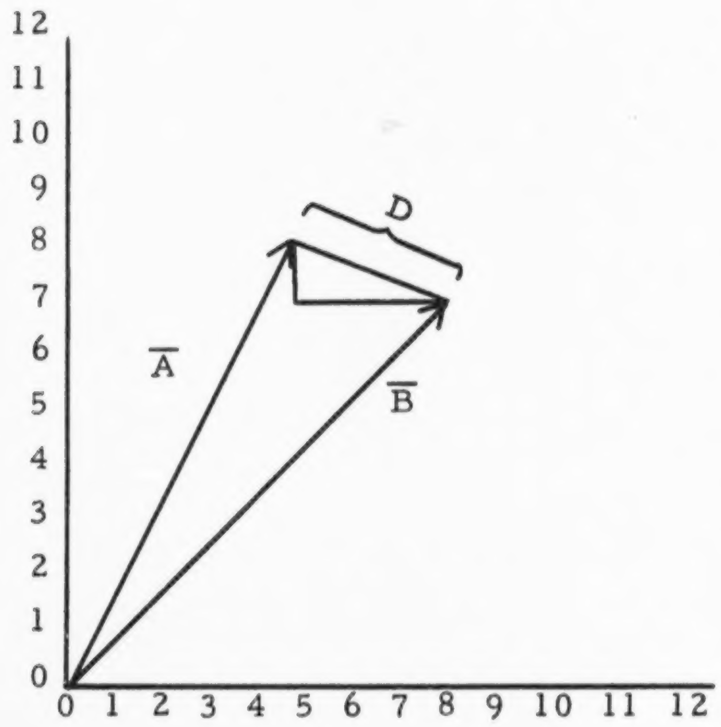

12
11
10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12
D
$\overline{A}$
$\overline{B}$

Figure III. 4. Two vectors in two dimensions and the computation of the Euclidean distance between them

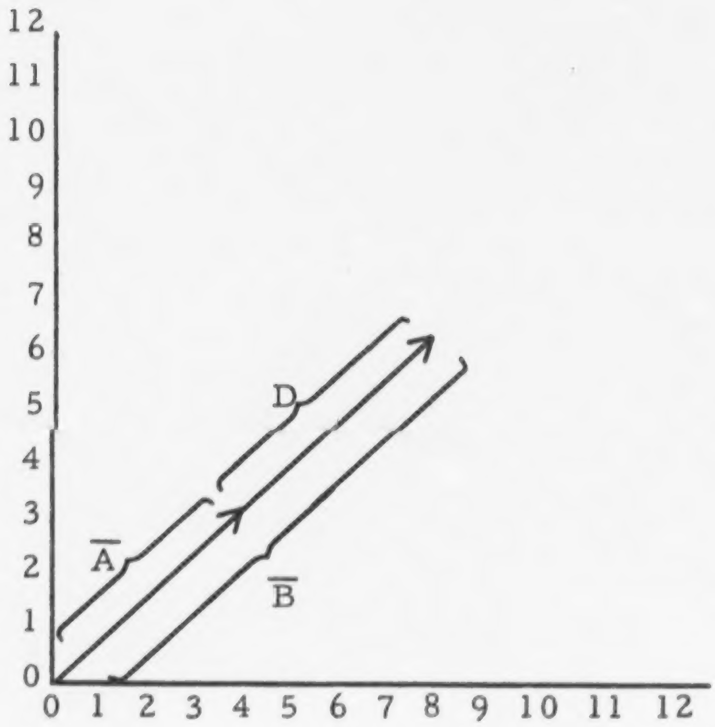

12
11
10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12
$\overline{A}$
D
$\overline{B}$

Figure. III. 5. Error caused by the computation of the Euclidean Distance between two vectors in two-dimensional space, when related by a scalar multiplication factor.

2. Tanimoto (overlap)

$$\text{Tanimoto} = \frac{\sum_{i=1}^{N} (d^i_1 \cdot d^i_2)}{\sum_{i=1}^{N} (d^i_1)^2 + \sum_{i=1}^{N} (d^i_2)^2 - \sum_{i=1}^{N} (d^i_1 \cdot d^i_2)^2}$$

The Tanimoto measure also produces a similarity measure normalized to the range from zero to unity.

III. 2 CONCORDANCE GENERATION

The first analysis tool of the Batch System is a two-level concordance to the entire data base. By "two-level" is meant a combination of concordances of stems for each document in the collection and the combined merge of the document concordances. In the jargon of the implementation the document concordances are called "micro-concordances," while the concordance to the entire collection is called the "master concordance." The formats of these individual data sets are described in detail in section V. 4 of this report.

The concordances produced during this first stage of data base analysis are not collections of raw token counts, but rather are the result of preliminary morphological analysis or stemming and a process of common-word elimination (stop words). By stemming and removing common words the relative size of the data base is reduced, and tokens are eliminated which in all probability have little chance of becoming keywords for retrieval in the On-Line System. In addition some progress is made toward grouping together of words with like meanings, especially those words which share a common root morpheme, such as "computer," "computing," "computation," *et cetera*.

Because the concordances are the input data to the statistical filter package, more than just the stem counts are collected. Data are gathered on the number of stems in each document as well, for use in normalizing the within-document frequency count. The concordances themselves are retained in the "micro-concordance" format for later use in "document concept vector" generation, the method of automatic document classification.

III.3 THE STATISTICAL FILTER PACKAGE

Once the data base has passed through the basic analysis of the concordance generation, the second phase is entered: separation of the "content stems" from the "non-content stems." The selection of content stems from the collection of raw stems is performed by the statistical filtering program, which collects those word stems most promising for the characterization of documents by an analysis of both the stem rank-frequency distribution and the variation of that distribution over the entire document collection. Details of the method chosen and implementation documentation are contained in section V.5 of this report.

Basically the statistical filter attempts to reject those stems which have a high frequency of occurrence and are evenly distributed across the collection. Because of certain characteristics of the statistical measure, stems are also rejected if they are very low in frequency as well (occur two or fewer times). Stems are rejected if they fall in the high or low areas of frequency range or have an even distribution across the document collection. Stems are in turn ranked by descending value of the filtration metric, and an arbitrary cut-off is set. In the case of this system, 5,000 of the highest ranking stems were retained, mostly because 5,000 stems represented the maximum amount of concordances which could be handled in core at once for the stem clustering algorithm chosen.

On the other hand, it is clear that as larger collections are analyzed using this method the number of content stems retained must be increased, else some of the documents fail to be characterized by any document concept vector at all (a "null" vector). In fact with the collection treated in this research 180 documents failed to have any content stems in them at all, indicating that 5,000 content stems was slightly too small a figure for a collection of approximately 36,000 documents. However, scheduling deadlines prevented an increase in this figure.

Since, in turn, the output of the statistical filter was the input for the next phase of operation, additional information was produced by the package. The final output of the filtration package was a collection of concordances, one for each content stem retained. However, because some of the concordances could reach quite large proportions, it was decided to sort these concordances by descending token counts, retaining the highest 30 document-weight pairs per content stem, again because of the requirement that as much as possible of the clustering data be manipulated in high-speed core storage for reasons of computational economy. The process of reducing the content stem concordances to a uniform maximum size was termed "dimension reduction," in the jargon of the project. There was some legitimate concern that valuable information would be lost for some of the stems, but, as it turned out, a relatively small number of content stems were affected by the reduction process, leaving the majority of the information intact.

III.4 CREATION OF THE CONCEPT DICTIONARY (KEYWORD LIST)

After the dimension-reduced concordances for the 5,000 content stems have been produced, there remains the task of mapping the content stems onto the 1500-dimensional hyperspace used to represent the different semantic domains present in the collection. This process is termed "stem clustering." Details of this process are presented in section V.6 of this report.

The clustering algorithm operates in three major phases:

1. All non-zero correlations are computed between all 5,000 reduced concordances, treating the document number value as the dimension of the vector component and the within document frequency (normalized) as the magnitude of the vector component. The file of correlation measures is sorted in descending order of the correlation metric.
2. Next at attempt is made to find those stems which as a group correlate strongly to other stems but not to those stems considered as cluster centers.
3. The last phase computes the correlation measure between each of the 5,000 content stems and the stems which were deemed "concept centers." The highest four concept-weight pairs are retained.

The collection of 5,000 sets of four concept-weight pairs is termed the concept dictionary and is used both in automatically classifying documents and in running the On-Line Retrieval System.

III.5 AUTOMATIC DOCUMENT CLASSIFICATION

The last phase of the Batch System involves the automatic classification of the document collection in terms of the concept space established during stem clustering. In order to accomplish this classification the system uses the concept dictionary developed during stem clustering and the micro-concordances from the first phase of operation. Output consists of two different types of information:

1) the document concept vectors themselves, i.e., the weighted mappings of the content stems contained in one document onto the vector space.
2) concept-document pairs which are used by the on-line system support package in preparing the inverted list pointers for retrieval of the document concept vectors during the On-Line System operation.

By sorting the second batch of information mentioned above, using the concept number as the sortkey, an inverted list is obtained. On-line support programs then compact the sparse list into a form more suitable for on-line storage and retrieval.

SECTION 4

SECTION IV

SYSTEM DESCRIPTION: THE ON-LINE SYSTEM

The section of the report is concerned with four major topics: comparison between the present and previous versions of the on-line system; motivation for the reprogramming effort; an overview of the present system; sample dialogues of the present system in operation, with two kinds of users. The four subsections cover each of the topics mentioned above.

IV.1 THE PREVIOUS ON-LINE SYSTEM AND THE PRESENT ONE COMPARED

The present on-line Dialogue Processor is literally the offspring of its predecessor system, prepared for Contract F30602-69-C-0038 and documented fully in RADC-TR-70-44. The reader is referred to this report for details of that implementation effort. However, this section outlines briefly the major points of the previous processor, as the present version is best understood in the light of the former version. Similarities between the two systems will be discussed first.

The following are points of identity or great similarity:

1. Makes available Terse and Verbose dialogue
2. Has Tutorial aids available
3. Usage is interactive (on-line system)
4. Analyzes free text input (in English) to produce the query.
5. Allows direct internal manipulation of the generated query, as well as direct query building (without any input text).
6. Document-document searching may be used
7. Retrieval is by either temporary I. D. or by actual accession number of a document.
8. The user is allowed direct control over options, as well as the default control by the system itself.

The following points are areas of great dissimilarity between the present system and its predecessor:

	Present System	Previous System
1.	coded in JOVIAL	coded in time-sharing FORTRAN
2.	uses conversation GEFRC for for terminal I/O	uses the time-sharing executive for terminal I/O
3.	has access to batch-world files on disk-pack as well as GECOS-III permfiles	is restricted to permanent files within GECOS-III
4.	utilizes all random-access files for the on-line files	sequential access only
5.	user messages are prestored internally	user messages are contained on an external sequential-access file
6.	occupies 29K of core in the batch-world	required 51K of core allocated to the time-sharing system
7.	system may be extended to access at least 40, 000 documents without changing core requirements	prototype system only -- runs on a collection of 50 documents
8.	is subject to swap-out for indeterminate lengths of time as is any batch program	by-and-large is not swapped out for any noticeable length of time
9.	introduces new options for the user, as well as allowing certain short-cuts to be taken; high degree of modularity	processor does not delineate the various functions by name; virtually all code is contained in one main program, several subroutines excepted
10.	packs internal tables tightly to conserve core space	table packing not available in FORTRAN
11.	system available in system-loadable form on a disk pack	system used in source form from permfiles only

The above points are by no means exhaustive in comparing the two on-line systems, as an actual point-by-point listing would require inordinate amounts of space and have little relevance to the discussion. The next subsection of this report discusses the motivation for reprogramming the previous system, advantages and disadvantages of JOVIAL for coding the processor, modularization of the system, and other pertinent points.

IV.2 MOTIVATION FOR REPROGRAMMING THE PREVIOUS SYSTEM

This subsection covers the motivation for reprogramming the TSS-FORTRAN version of the on-line Dialogue Processor. The decision to recode was not approached lightly, but was the result of much weighing of pros and cons. Of primary consideration was the matter of dialogue speed, but other factors such as expandability and adaptability to other data bases weighed heavily also. In addition, the matter of access to removable disk-packs was of considerable concern, as the amount of space available to any one user within the permanent file system was limited to a small fraction of the file space required to contain the on-line data base.

IV.2.1 Speed of Operation

Since the retrieval system is indeed on-line, the matter of response time is all important and was upper most in the minds of the designers when considering modifications to the previous version. Response time in the on-line system is related to two factors; batch or time-sharing usage; user load on the system while the retrieval system is in use.

It is clear that time-sharing users obtain fewer CPU cycles per unit of elapsed time for the following reason: not only does the TSS user compete with other TSS users for available cycles, but the time-sharing executive itself is just another batch program. Admittedly, TSS runs at a considerably higher urgency level than most batch programs, but there is nothing to prevent the machine operator from raising the urgency of any batch job at will. In turn, the TSS executive must compete with other batch jobs for CPU cycles and peripheral resource allocation. There is an additional matter: core space.

The time-sharing executive occupies a portion of the core allocated to the time-sharing system, the remaining core being allocated to all the TSS users. Thus each TSS user competes with all other TSS users for core space within TSS itself. The executive meets these demands on the limited amount of core by a process of constant swap-out/roll-in, eating up yet more elapsed time. Running the on-line system directly in the batch world avoids the overhead inherent in the operations described above. Especially during the actual retrieval process, the on-line system no longer is I/O bound by the terminal and uses a much greater amount of CPU time than when conversing directly with the user. Avoiding the time-sharing overhead greatly quickens the response time, especially considering the fact that the present system operates on a much larger data base than does the previous one.

IV. 2. 2 Core Requirements Problems

As was briefly touched upon above, a program using the TSS executive is limited by the TSS allocation in the amount of core it can use. Especially large programs, such as the previous Dialogue Processor, require advance arrangements for additional time-sharing system core. The same is not true of batch programs. Jobs larger than a predetermined limit may require operator intervention to be run, but special advance arrangements are not necessary. However, this matter does not come up with the present processor, as its core requirements are considerably lighter than the previous processor.

Through the scheme of overlays, the new Dialogue Processor requires only 29K of core, not a particularly large allocation. This economy of core is not only the result of the overlay scheme, but is also related to the use of JOVIAL as the programming language. The choice of JOVIAL will be discussed below. The last significant reason for lighter core requirements is inherent in the difference between batch world executable and TSS generated machine code.

Briefly, the significant difference between TSS and batch code is the method of address computation. Normal batch code uses the address portion of an instruction to reference the data desired, or at least a combination of the address field and the contents of a register. TSS code, on the other hand, uses the instruction counter as a modifier for all relocatable address fetches, with the result that addressing of operands is a good bit less economical. This difference in type of generated code results in proportionally more instructions in TSS code than in batch code thus increasing the length of programs. Therefore, batch programs are more economical of core space.

IV. 2. 3 Access to Large On-Line Files

One of the requirements of the on-line system is that it be able to access rather large, random access files of on-line data. Under the TSS system, however, all files referenced must reside within the permanent file system, batch world files being excluded. If a batch on-line program is used, however, all files within GECOS-III may be accessed, including magnetic tape, mass storage (fixed head disk or drum), and disk-packs. In the case of this system all the on-line large data files are organized on removable disk packs, as well as all of the operational programs and links. Because the revised system is basically a batch world program, it has access to all the above named types of data files.

IV. 2. 4 Ease of Modularization and Overlay Structure

From the earliest moment when redesign and reprogramming were considered, it was clear that the predecessor system not only was too unitized in structure but also that it required too much core to be practical. By breaking up the processor code into user functions and further into actual overlays, the overall system design was considerably strengthened. Now changes could be made to portions of the system without affecting those sections already completed and operational, an absolute necessity for a program as large as the Dialogue Processor. In addition, the future direction of the whole on-line system was changed. (Section VI. 6 goes into this matter in more depth.)

The matter of convenience is not to be overlooked. Since the processor is indeed a batch-world program (as discussed in IV. 2. 3), it is not necessary to have the on-line data base remain on permanent files. Two advantages thereby accrue: dismounting the on-line files when they are not in use decreases the chances of accidental file destruction; the system becomes portable, easily installed in a minimum of time and effort on any compatible computing system (HISI 615/625/635 or 645 with the simulator package). In addition, the random-access space taken up by the files is not dedicated to this Dialogue Processor, as other disk packs may be mounted on the drives when the on-line system is not in use.

The chief disadvantage of all removable files is that only one user may experiment with the system at a time. If all the files were stored within the permanent files system, several users could access the common files at once (as long as they were used with READ permission only). However, the advantages of the removable files far out weighs the single advantage of permanent files.

IV.2.5 Language of the Reprogramming Effort

Once the decision had been made to reprogram the on-line system, there remained the choice of programming language. It would have seemed that retaining FORTRAN (batch version) for the system should have had some distinct advantages. However, many of the problems inherent in the previous system stemmed directly from the use of FORTRAN. Therefore, when the decision was made, JOVIAL was chosen instead. The reasons for this choice are outlined below.

In the first place one of the weaknesses of the predecessor system was its uneconomical use of available storage; table packing is simply not possible in straight FORTRAN without resort to assembly-language subroutines. In addition there are no built-in mechanisms in FORTRAN for handling of character data, an absolute essential in this system. The JOVIAL language contains all these features mentioned above: sophisticated table packing and character handling. Besides, JOVIAL syntax allows most of all the HISI assembly language instructions to be coded in line with the regular JOVIAL code through the inclusion of the "left" and "right" brackets DIRECT and JOVIAL.

Additional reasons for using JOVIAL rather than FORTRAN are among the following:

1. Reduce coding time by using previously debugged code taken from portions of the batch system.
2. Insure identical treatment of stem analysis by using the analysis routines from the micro-concordance generator.
3. Insure identity of file structures as generated by the batch portion of the system.
4. Increase operational speed of the system through more efficient byte manipulation.
5. Decrease the core requirements.

IV.3 PRESENT SYSTEM STATUS

The current implementation of the on-line system has been completed to the point where it is functionally and operationally complete; retrieval queries may be entered; documents and bibliographic data retrieved; queries modified. The system is easily expanded from the demonstration system up to the full data base with a very minimum of effort, or reduced to a skeletal system printing no more than the accession number of a retrieved document. Response times are rapid, and persons having no familiarity with data processing easily adapt to the system. The tutorial portions of the system allow a "raw" user to learn about the system as he has the time and inclination to explore.

A detailed user's manual is contained in Appendix B of this report. Section IV.4 of this report contains several commented man-machine dialogues with the system, one as might be expected from a new user, the others from more experienced researchers.

IV.4 SAMPLE MAN-MACHINE DIALOGUES WITH RUNNING COMMENTARY

This subsection contains two examples of on-line usage, the first between an inexperienced user (tyro) and the system, the second showing dialogue between an experienced user and the system. Running commentary has been included to aid the reader in understanding just exactly what is happening at each point in the dialogue. Examples of retrieved bibliographies and abstracts have been truncated to conserve space on the printed page. Each sample dialogue is substantially complete, including sign-on to the time-sharing system, initiation of the on-line retrieval system, and sign-off from the time-sharing system.

IV.4.1 Man-Machine Dialogue: The "Tyro" User

In this subsection the all upper case text represents actual output from the user teletypewriter; commentary is indented and smaller in size. User responses to system questions always appear after the character sequence 05? .

```
***   GE-TSS N0TIFIED   ***

THIS IS THE GE-600 TSS SYSTEM 0N  05/03/71 AT 13.284  CHANNEL 0106


IISER ID -BIDBZTAB>77
PASSW0RD--
[redacted]
18 BL0CKS FILE SPACE AVAILABLE
```

The user signs onto the time-sharing system.

```
 SYSTEM ?CARD
```

CARDIN is entered to initiate running of the processor.

```
0LD 0R NEW-0LD RUNDIA$SUCCESS
READY
```

File name is RUNDIA with a password of SUCCESS.

```
*RUN
  SNUMB # 1051T
```

Request initiation of the job.

```
 CARD F0RMAT,DISP0SITI0N ?
 N0RM(;),TALK
```

Response indicates normal tab positions with a tab character of ";." Disposition is TALK, i.e., a direct-access-connection is to be established. The user now waits until the on-line retrieval system is activated.

```
R.A.D.C. 0N-LINE D0CUMENT RETRIEVAL SYSTEM "SIGNING 0N."
 ABSTRACTS & BIBLI0GRAPHIES ARE C0PYRIGHT 1967-1970
 AMERICAN PSYCH0L0GICAL ASS0CIATI0N. REPR0DUCTI0N PR0HIBITED.
  PLEASE AWAIT FURTHER INSTRUCTI0NS FR0M THE SYSTEM. THANKS.
```

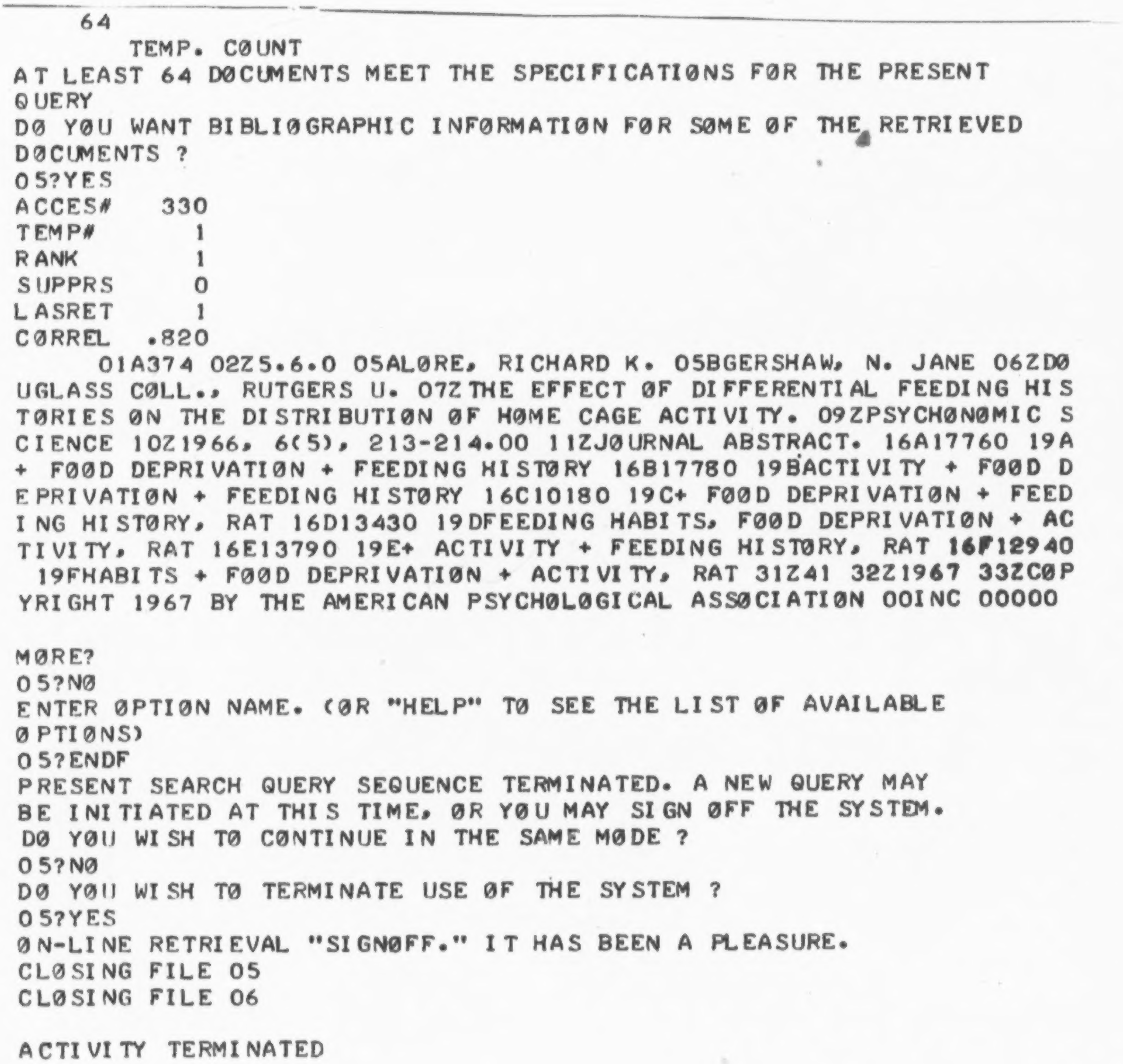
The retrieval system has now started into operation; the user must follow its instructions to continue onward.

```
 IS N0RMAL SYSTEM 0PERATI0N DESIRED ?
 05?YES
```

The request for NORMAL OPERATION is the dividing point between the "tyro" user and the sophisticated experimenter. A "YES" response carries the user directly into creating a query.

```
ENTER W0RDS (FREE TEXT) F0R INITIAL SEARCH QUERY, F0LL0WED
BY "EXCLAMATI0N-P0INT"
05?F00D DEPRIVATI0N !
***ANALYZING QUERY***
***QUERY ANALYSIS C0MPLETED***
***QUERY-D0CUMENT C0RRELATI0N BEGUN***
```

The retrieval system keeps the user up to date on the progress it has made in meeting the retrieval specifications.
At completion of a retrieval, the system engages the user in a dialogue about the retrieved documents, printing them out as the user instructs. In this case, the user obtains the bibliography for one document.



```
     64
        TEMP. C0UNT
AT LEAST 64 D0CUMENTS MEET THE SPECIFICATI0NS F0R THE PRESENT
QUERY
D0 Y0U WANT BIBLI0GRAPHIC INF0RMATI0N F0R S0ME 0F THE RETRIEVED
D0CUMENTS ?
05?YES
ACCES#    330
TEMP#       1
RANK        1
SUPPRS      0
LASRET      1
C0RREL  .820
     01A374 02Z5.6.0 05AL0RE, RICHARD K. 05BGERSHAW, N. JANE 06ZD0
UGLASS C0LL., RUTGERS U. 07ZTHE EFFECT 0F DIFFERENTIAL FEEDING HIS
T0RIES 0N THE DISTRIBUTI0N 0F H0ME CAGE ACTIVITY. 09ZPSYCH0N0MIC S
CIENCE 10Z1966, 6(5), 213-214.00 11ZJ0URNAL ABSTRACT. 16A17760 19A
+ F00D DEPRIVATI0N + FEEDING HIST0RY 16B17780 19BACTIVITY + F00D D
EPRIVATI0N + FEEDING HIST0RY 16C10180 19C+ F00D DEPRIVATI0N + FEED
ING HIST0RY, RAT 16D13430 19DFEEDING HABITS, F00D DEPRIVATI0N + AC
TIVITY, RAT 16E13790 19E+ ACTIVITY + FEEDING HIST0RY, RAT 16F12940
 19FHABITS + F00D DEPRIVATI0N + ACTIVITY, RAT 31Z41 32Z1967 33ZC0P
YRIGHT 1967 BY THE AMERICAN PSYCH0L0GICAL ASS0CIATI0N 00INC 00000

M0RE?
05?N0
ENTER 0PTI0N NAME. (0R "HELP" T0 SEE THE LIST 0F AVAILABLE
0PTI0NS)
05?ENDF
PRESENT SEARCH QUERY SEQUENCE TERMINATED. A NEW QUERY MAY
BE INITIATED AT THIS TIME, 0R Y0U MAY SIGN 0FF THE SYSTEM.
D0 Y0U WISH T0 C0NTINUE IN THE SAME M0DE ?
05?N0
D0 Y0U WISH T0 TERMINATE USE 0F THE SYSTEM ?
05?YES
0N-LINE RETRIEVAL "SIGN0FF." IT HAS BEEN A PLEASURE.
CL0SING FILE 05
CL0SING FILE 06

ACTIVITY TERMINATED
```

After having obtained all the information he desired, the user signs off from the retrieval system. He is returned to the time-sharing system "build mode."

IV.4.2 Man-Machine Dialogues: Experienced Users

This subsection contains a complete dialogue between the system and a sophisticated user, as well as selected portions of other dialogues to illustrate some of the power available to a searcher once he becomes familiar with details of operation.

IV.4.2.1 Complete Sophisticated Dialogue (Without Time-Sharing System Sign-On).

```
R.A.D.C. 0N-LINE D0CUMENT RETRIEVAL SYSTEM "SIGNING 0N."
 ABSTRACTS & BIBLI0GRAPHIES ARE C0PYRIGHT 1967-1970
 AMERICAN PSYCH0L0GICAL ASS0CIATI0N. REPR0DUCTI0N PR0HIBITED.
 PLEASE AWAIT FURTHER INSTRUCTI0NS FR0M THE SYSTEM. THANKS.
IS N0RMAL SYSTEM 0PERATI0N DESIRED ?
05?N0
D0 Y0U WANT AN EXPLANATI0N 0F THE AVAILABLE M0DES ?
05?N0
"M0DE" FLAGS ARE ALL 0FF. IDENTIFY NUMBERS 0F FLAGS T0 BE SET
"0N", F0LL0WED BY A "ZER0" VALUE.  VALUES ARE C0MMA DELIMITED
05?1,3,4,5,0
N0RMAL 0PERATI0N ?
05?YES
```

The system sign-on is the same as for the "tyro" user, except that now "normal" system operation is refused. Instead "mode" flags are set allowing more control over system operation. Once this has been accomplished, the user returns to the same point as before. Note that Terse dialogue is now in use.

```
ENTER INITIAL QUERY W0RDS
05?F00D AND S0CIAL AND L0VE DEPRIVATI0N !
PRINT QUERY W0RDS ?
05?N0
***ANALYZING QUERY***
***QUERY ANALYSIS C0MPLETED***
***QUERY-D0CUMENT C0RRELATI0N BEGUN***
```

Now a query is entered, but since mode flag 3 (print query information) has been turned on, the user could obtain a list of the query words retained and their stems. He decides not to look at these tables for the moment. The retrieval continues in the normal fashion.

```
     64
        TEMP. C0UNT
 N0. 0F HITS >= 64
 PRINT "TEMP" TABLE ?
 05?N0
 PRINT BIBLI0GRAPHY ?
 05?N0
```

The "TEMP" table contains 64 entries at retrieval termination, and at least 64 documents were retrieved. Since mode flag 4 was turned on (print retrieval information), the user is queried by the system to see if he desires this information before continuing. He elects not to see "TEMP" for the moment, nor does he wish to see any bibliographies.

```
 ??
 05? SKIM
 05?C0000,W0.480
       9 ENTRIES IN "TEMP"
```

Instead he chooses to employ a powerful "skimming" function. A minimum correlation value is set for the "TEMP" table, and those current entries below this value are purged from the table. Ordinarily this function would not be used until some familiarity is obtained with a particular query. But in this case the user had at some previous session examined portions of the documents below the cutoff of 0.480 and had decided that they were not very relevant to his search. After the "SKIM" is performed, the user is informed of the new current size of "TEMP."

```
??
05?CHAT
PRINT "TEMP" TABLE ?
05?YES
 ACCESSI0N   TEMP0RARY    C0RRELATI0N         RANK     BIB.DATA PRINTED/
   NUMBER     IDENT.      (WHEN LAST RETRIEVED)        RET'VD LAST QUERY
      1         1            .539               1         0    /     1
    1966        2            .533               2         0    /     1
     738        3            .517               3         0    /     1
    1336        4            .517               4         0    /     1
     538        5            .504               5         0    /     1
     330        6            .494               6         0    /     1
M0RE?
05?YES
     998        7            .492               7         0    /     1
     373        8            .488               8         0    /     1
    1506        9            .486               9         0    /     1
THAT'S "ALL"
```

Through the use of CHAT the search now returns to the same point which he reached before: arequest to print retrieval information. Having employed "SKIM," he now verifies that only nine items remain in "TEMP." Note that the print-outs of the "TEMP" information are batched, so that the user has better control over the amount of material typed at his terminal.

```
PRINT BIBLI0GRAPHY ?
05?0PT
??
05?CLR
??
05?DDC
SPECIFY D0CS.
05?A1
DIRECT C0N. VECT. ACTI0N ?
05?N0
```

Nor does the user want any bibliographic information. Instead he clears the present query concept vector in preparation for document-document searching. Then the document concept vector for document number 1 (an accession number of 1) is used instead of a query derived from text-word analysis. The user elects not to modify the query vector directly, and control passes to the retrieval.

```
RETRIEVE ?
05?YES
EXCLUDE PREVI0US D0CS.?
05?YES
SUPPRESS PREV. PRINTED ?
05?N0
```

Before allowing a retrieval to take place, the system queries the user about two possible other conditions. The first possibility concerns excluding already retrieved documents from re-retrieval. The second decision point permits the user to turn off all the "suppress" flags in "TEMP." Once these flags are off, bibliographic information already printed may be printed again.

```
   10
TEMP. ID. STARTS WITH NUMBER --
  247
      SPACES IN "TEMP." WANT M0RE ?
05?N0
```

Showing a "TEMP" contents of 10, the system asks the user if more space is desired for retrieved documents. Since only 10 spaces are used, the uses answers NO. The remaining dialogue shows the document-document search query completed, along with a dump-out of a large portion of the "TEMP" table. Note that the previously-retrieved documents are not re-retrieved, but their entries remain in the table, marked as not having been retrieved on the last go-around.

```
***ANALYZING QUERY***
***QUERY ANALYSIS C0MPLETED***
***QUERY-D0CUMENT C0RRELATI0N BEGUN***
     73
        TEMP. C0UNT
N0. 0F HITS >= 64
PRINT "TEMP" TABLE ?
05?YES
  ACCESSI0N   TEMP0RARY    C0RRELATI0N          RANK    BIB.DATA PRINTED/
    NUMBER      IDENT.    (WHEN LAST RETRIEVED)         RET'VD LAST QUERY
     1449         10           .865                1         0   /    1
     1506         11           .836                2         0   /    1
      443         12           .803                3         0   /    1
     1213         13           .785                4         0   /    1
      449         14           .783                5         0   /    1
     1406         15           .783                6         0   /    1
M0RE?
05?YES
     1407         16           .781                7         0   /    1
      585         17           .778                8         0   /    1
     1969         18           .767                9         0   /    1
     1161         19           .766               10         0   /    1
      547         20           .761               11         0   /    1
M0RE?
05?YES
     1976         21           .758               12         0   /    1
      426         22           .745               13         0   /    1
      719         23           .739               14         0   /    1
      203         24           .707               15         0   /    1
     1318         25           .694               16         0   /    1
M0RE?
05?YES
```

```
     448          26          .685          17        0   /   1
    1816          27          .683          18        0   /   1
     561          28          .674          19        0   /   1
    1948          29          .659          20        0   /   1
     770          30          .647          21        0   /   1
MØRE?
05?YES
    1220          31          .633          22        0   /   1
    1462          32          .632          23        0   /   1
     416          33          .629          24        0   /   1
    1533          34          .627          25        0   /   1
      29          35          .623          26        0   /   1
MØRE?
05?YES
    1964          36          .613          27        0   /   1
    1151          37          .610          28        0   /   1
    1349          38          .609          29        0   /   1
    1297          39          .607          30        0   /   1
     546          40          .604          31        0   /   1
MØRE?
05?YES
     499          41          .600          32        0   /   1
    1977          42          .597          33        0   /   1
    1590          43          .596          34        0   /   1
    1274          44          .595          35        0   /   1
     895          45          .592          36        0   /   1
MØRE?
05?YES
    1337          46          .589          37        0   /   1
    1177          47          .585          38        0   /   1
    1963          48          .577          39        0   /   1
    1518          49          .577          40        0   /   1
    1143          50          .576          41        0   /   1
MØRE?
05?YES
     511          51          .573          42        0   /   1
    1915          52          .570          43        0   /   1
    1386          53          .570          44        0   /   1
    1260          54          .562          45        0   /   1
     465          55          .559          46        0   /   1
MØRE?
05?YES
     789          56          .545          47        0   /   1
     694          57          .541          48        0   /   1
    1882          58          .541          49        0   /   1
     629          59          .540          50        0   /   1
       1           1          .539           1        0   /   0
```

```
M0RE?
05?YES
     258        60          .539        51      0  /  1
    1405        61          .535        52      0  /  1
     425        62          .534        53      0  /  1
    1966         2          .533         2      0  /  0
    1707        63          .531        54      0  /  1
M0RE?
05?YES
    1421        64          .530        55      0  /  1
    1983        65          .527        56      0  /  1
     478        66          .526        57      0  /  1
     951        67          .526        58      0  /  1
    1146        68          .526        59      0  /  1
M0RE?
05?YES
     738         3          .517         3      0  /  0
    1336         4          .517         4      0  /  0
    1876        69          .516        60      0  /  1
    1154        70          .510        61      0  /  1
      15        71          .505        62      0  /  1
M0RE?
05?0PT
??
05?ENDRUN
SYSTEM L0GGING "0FF" .  IT HAS BEEN A PLEASURE W0RKING WITH Y0U.
CL0SING FILE 05
CL0SING FILE 06

ACTIVITY TERMINATED
```

IV.4.2.2 Dialogue Portions Illustrating Some of the System Features. This sub-section is a collection of sections of various dialogues from different retrieval sessions. Each portion is intended to illustrate one or more of the optional features available to the sophisticated user. The dialogue starts in verbose mode.

In the following example the user asks for an explanation of the available modes of operation and sets some of them.

BLANK PAGE

```
D0 Y0U WANT AN EXPLANATI0N 0F THE AVAILABLE M0DES ?
05?YES
M0DES ARE N0RMALLY "0FF," BUT CAN BE TURNED "0N" BY TYPING IN
NUMBER 0R SEQ. 0F NUMBERS (1,3,5,0).  AVAILABLE M0DES F0LL0W:
 FLAG NUMBER          ACTI0N
       1              SELECT "TERSE" DIAL0GUE F0RM.
       2              SKIP INITIAL QUERY F0RMATI0N FR0M W0RDS
       3              MAKE AVAILABLE - QUERY W0RDS, STEMS, AND
                      C0NCEPTS, BEF0RE RETRIEVAL TAKES PLACE
       4              "TEMP" TABLE AVAILABLE AFTER RETRIEVAL
       5              ALL AVAILABLE 0PTI0NS MAY BE USED
"M0DE" FLAGS ARE ALL 0FF. IDENTIFY NUMBERS 0F FLAGS T0 BE SET
"0N", F0LL0WED BY A "ZER0" VALUE.  VALUES ARE C0MMA DELIMITED
05?3,4,5,0
```

This next sample shows the use of HELP to list the available options. Since mode 5 was on, all options were explained.

```
ENTER 0PTI0N NAME. (0R "HELP" T0 SEE THE LIST 0F AVAILABLE
0PTI0NS)
05?HELP
AVAILABLE 0PTI0NS F0LL0W:
 "ENDF" - TERMINATE THIS SEARCH QUERY SEQUENCE IN 0RDER T0
          START A NEW SEQUENCE 0R SIGN 0FF THE SYSTEM.
 "M0D"  - M0DIFY 0R REPLACE THE PRESENT QUERY AND C0NTINUE
          THE PRESENT QUERY SEQUENCE.
 "D0C"  - PRINT DATA F0R D0CUMENTS RETRIEVED DURING THIS
          SEQUENCE 0R ANY D0CUMENTS 0F KN0WN ACCESSI0N NUMBER.

 "0FF"  - CREATE FILE F0R 0FFLINE D0CUMENT PRINTING
 "CHG"  - CHANGE M0DE 0F 0PERATI0N.(QUERY N0T ENDED)
 "C0N"  - INSPECT THE C0NCEPT VECT0RS 0F D0CUMENTS
 "RET"  - EXECUTE THE PRESENT RETRIEVAL REQUEST.
 "DEL"  - DELETE UNWANTED D0CUMENTS JUST RETRIEVED
 "SEE"  - INSPECT THE EXISTING QUERY
 "CLR"  - ERASE THE EXISTING QUERY
 "WRD"  - ADD 0R DELETE QUERY W0RDS
 "DDC"  - PERF0RM D0CUMENT-D0CUMENT C0RRELATI0N
 "WGT"  - MANIPULATE QUERY C0NCEPT VECT0RS DIRECTLY
ENTER 0PTI0N NAME. (0R "HELP" T0 SEE THE LIST 0F AVAILABLE
0PTI0NS)
```

Function SKIM is used below to purge all items from the TEMP table with correlations below 0.525.

```
ENTER OPTION NAME. (OR "HELP" TO SEE THE LIST OF AVAILABLE
OPTIONS)
05?SKIM
05?C0000,W0.525
     6 ENTRIES IN "TEMP"
ENTER OPTION NAME. (OR "HELP" TO SEE THE LIST OF AVAILABLE
OPTIONS)
```

An example of the use of the function WRD, when the SEE option has previously been used, thus printing the contents of the current query concept vector.

```
05?WRD
QUERY WORDS ARE:
DEPRIVATION
FOOD
LOVE
SOCIAL
PRESENT QUERY
CON-  WEIGHT
CEPT
  108     .133
  253     . 72
  362     .108
  422     .218
  428     .231
  560     .177
  572    1.000  ◄──────── Cluster center
  610     .106
  633     .128
  646     . 58
  652     .171
  662     .225
  912     .199
  914     .207
  934     .209
  950     .122
  986     .218
 1042    1.000  ◄──────── Cluster center
 1105     . 63
 1193     . 75
 1197     .123
 1297     . 74
 1364     .150
 1394    1.000  ◄──────── Cluster center
??
```

The following shows manipulation of the words forming a query, including replacement of the current query, manual clearing of the query concept vector through the use of the CLR function, and a manual build of the query concept vector followed by a dump out of its contents.

```
REPLACE ALL ØF PRESENT QUERY ?
05?YEA
ENTER WØRDS
05?DEPRIVATIØN !
DELETE WØRDS ?
05?WRD ◄──────────────────── Note the mode 5 use of an option
QUERY WØRDS ARE:              in place of a yes-no reply
DEPRIVATIØN
PRESENT QUERY
CØN- WEIGHT
CEPT
 108    .133
 253    . 72
 428    .231
 572   1.000
 610    .106
 633    .128
 646    . 58
 652    .171
 662    .225
 950    .122
1042   1.000
1105    . 63
1193    . 75
1297    . 74
1364    ,150
1394   1.000
??
05?CLR
??
05?BLD
QUERY-CØNCEPT-VECTØR BUILT....
??
05?WGT
PRESENT QUERY
CØN- WEIGHT
CEPT
 572   1.000
 633    .128
1193    . 75
1297    . 74
??
```

In the next example the function DOC is used to obtain copies of any document of known accession number. Note that no TEMP information is given for these documents, since they were not retrieved during the current retrieval session.

```
05?DØC

05?A1478
11.0.0 05AMASSUCCØ CØSTA, A. 05BPELUFFØ, N. 05CFERRARIS, A. 05DBIA
NCØ, B. 07ZANALISI PSICØ-SØCIØLØGICA DI FØRZE DEL LAVØRØ FEMMINILE
. 08ZPSYCHØSØCIØLØGICAL ANALYSIS ØF STRENGTHS IN FEMININE WØRK. 09
ZRIVISTA DI PSICØLØGIA SØCIALE E ARCHIVIØ ITALIANØ DI PSICØLØGIA G
ENERALE E DEL LAVØRØ 10Z1966, 33(2-3), 115-162. 00011ZL. L'ABATE.
16A16950 19AFEMALE WØRKER ADVANTAGES 16B20770 19BFEMALE WØRKER ADV
ANTAGES 16C18760 19CFEMALE WØRKER ADVANTAGES 31Z41 32Z1967 33ZCØPY
RIGHT 1967 BY THE AMERICAN PSYCHØLØGICAL ASSØCIATIØN I000NC 00000

PRINT ABSTRACT ?
05?YES
THERE ARE NØ ØBJECTIVE CRITERIA TØ EXCLUDE WØMEN FRØM SPECIALIZED
INDUSTRIAL JØBS. A GREATER DEGREE ØF EDUCATIØN ALLØWS WØMEN TØ BEC
ØME MØRE INDEPENDENT AND TØ WØRK MØRE SATISFACTØRILY. 00000
CØNTINUE ?
05?YES

05?A1675
     01A2389 02Z3.28.0 05ASTRØNG, PASCHAL N., JR. 06ZU. SØUTH FLØR
IDA 07ZCØMPARATIVE STUDIES IN SIMPLE ØDDITY LEARNING  II. CHILDREN
, ADULTS AND SENILES. 09ZPSYCHØNØMIC SCIENCE 10Z1966, 6(10), 459-4
60. 1100000ZJØURNAL ABSTRACT. 16A15230 19AØDDITY LEARNING, CHILDRE
N + ADULTS + SENILES 16B11610 19BØDDITY LEARNING, CHILDREN + ADULT
S + SENILES 16C10290 19CØDDITY LEARNING, CHILDREN + ADULTS + SENIL
ES 16D13965 19DØDDITY LEARNING, CHILDREN + ADULTS + SENILES 31Z41.
32Z1967 33ZCØPYRIGHT 1967 BY THE AMERICAN PSYCHØLØGICAL ASSØCIATIØ
N INC
PRINT ABSTRACT ?
05?YES
113 HUMANS FRØM THE AGE ØF 3.0 TØ HØSPITALIZED SENILE PATIENTS WER
E RUN ØN IDENTICAL 1 TRIAL, 2 PØSITIØN ØDDITY PRØBLEMS USED PREVIØ
USLY WITH CATS, RACCØØNS, MØNKEYS, AND CHIMPANZEES. THE YØUNGEST C
HILD TØ LEARN WAS 3 YR., 4 MØ. ØF AGE. 64  ØF THE 6-YR-ØLDS LEARNE
D WHILE ALL 12-YR-ØLDS AND CØLLEGE STUDENTS REACHED CRITERIØN PERF
ØRMANCE WITHIN THE 2ND DAY. SØME SENILES LEARNED THE PRØBLEM BUT S
HØWED ATYPICAL, INSIGHT-LIKE, LEARNING CURVES.  00000
CØNTINUE ?
05?YES
```

Document-document searching is illustrated in this next example, complete with a dump-out of a portion of the TEMP table showing the document used as the source of correlation with a correlation to itself of 1.000, i.e., identity.

```
??
05?DDC
SPECIFY D0CS.
05?A1273
DIRECT C0N. VECT. ACTI0N ?
05?N0
RETRIEVE ?
05?YES
TEMP. FILE INITIALLY EMPTY
EXCLUDE PREVI0US D0CS.?
05?N0
SUPPRESS PREV. PRINTED ?
05?N0
TEMP. FILE INITIALLY EMPTY
TEMP. ID. STARTS WITH NUMBER --
     1
***ANALYZING QUERY***
***QUERY ANALYSIS C0MPLETED***
***QUERY-D0CUMENT C0RRELATI0N BEGUN***
    64
       TEMP. C0UNT
N0. 0F HITS >= 64
PRINT "TEMP" TABLE ?
05?YES
 ACCESSI0N  TEMP0RARY   C0RRELATI0N        RANK   BIB.DATA PRINTED/
   NUMBER     IDENT.    (WHEN LAST RETRIEVED)     RET'VD LAST QUERY
    1273        1          1.000             1       0   /    1
     573        2           .619             2       0   /    1
     575        3           .488             3       0   /    1
    1488        4           .441             4       0   /    1
    1805        5           .433             5       0   /    1
     637        6           .400             6       0   /    1
M0RE?
05?YES
    1382        7           .399             7       0   /    1
    1235        8           .397             8       0   /    1
     580        9           .390             9       0   /    1
     124       10           .380            10       0   /    1
     576       11           .374            11       0   /    1
M0RE?
05?N0
```

This last example is an extended illustration of the use of function MOD. At first the user gets into a loop by refusing to perform any actions allowed by MOD. In this case he is led right back to where he started. Then, words are added to the current query, each time returning to WRD to examine the current content of the query words table.

```
05?MØD
DØC. - DØC. RETRIEVAL ?
05?NØ
REPLACE ALL ØF PRESENT QUERY ?
05?NØ
QUERY WØRD ACTIØN ?
05?NØ
DIRECT CØN. VECT. ACTIØN ?
05?NØ
RETRIEVE ?
05?NØ
DØC. - DØC. RETRIEVAL ?
05?NØ
REPLACE ALL ØF PRESENT QUERY ?
05?NØ
QUERY WØRD ACTIØN ?
05?YES
QUERY WØRDS ARE:
PATIENTS
THERAPY
ADD WØRDS ?
05?YES
ENTER WØRDS
05?DRUGS !
DELETE WØRDS ?
05?WRD
QUERY WØRDS ARE:
DRUGS
PATIENTS
THERAPY
ADD WØRDS ?
05?YES
ENTER WØRDS
05?DRUGS DRUGS SENILITY SENILITY SENILE !
DELETE WØRDS ?
```

```
0 5?WRD
QUERY W0RDS ARE:
DRUGS
DRUGS
DRUGS
PATIENTS
SENILITY
SENILITY
SENILE
THERAPY
ADD W0RDS ?
0 5?N0
```

SECTION 5

SECTION V

THE BATCH SYSTEM
IMPLEMENTATION DETAILS AND EXPERIMENTAL RESULTS

V.1 INTRODUCTION, SYSTEM OVERVIEW, AND FLOW OF OPERATION

This section details the implementation of the Batch System, briefly outlined in Section III, including experimental results, flowcharts, program listings, and details of the actual production. Topics covered include the following:

1) Code conversion of the data base
2) Data re-formatting (separation of the abstracts and bibliographies)
3) Concordance generation
4) Statistical filter operation
5) Concept-dictionary generation
6) Document concept vector generation (automatic document characterization)

The section terminates with a summary of the experimental results, suggestions for future work, and a detailed system production process chart. Figure V.1 illustrates the overall flow of data through the batch system.

V.2 CODE CONVERSION OF THE ORIGINAL SOURCE DOCUMENTS

V.2.1 Introduction

The data base for this project was a subset of a collection of machine-readable Psychological Abstracts, retained at the Rome Air Development Center. The tapes themselves were in IBM S/360, 9-track standard label format using the APL 8-bit BCD character codes. The first file of each tape is a standard file header label containing three, 20-word records, followed by a data file containing variable length physical records with a maximum block size of 32,760 characters. The third file is a standard trailer label consisting of two, 2-word records. Each

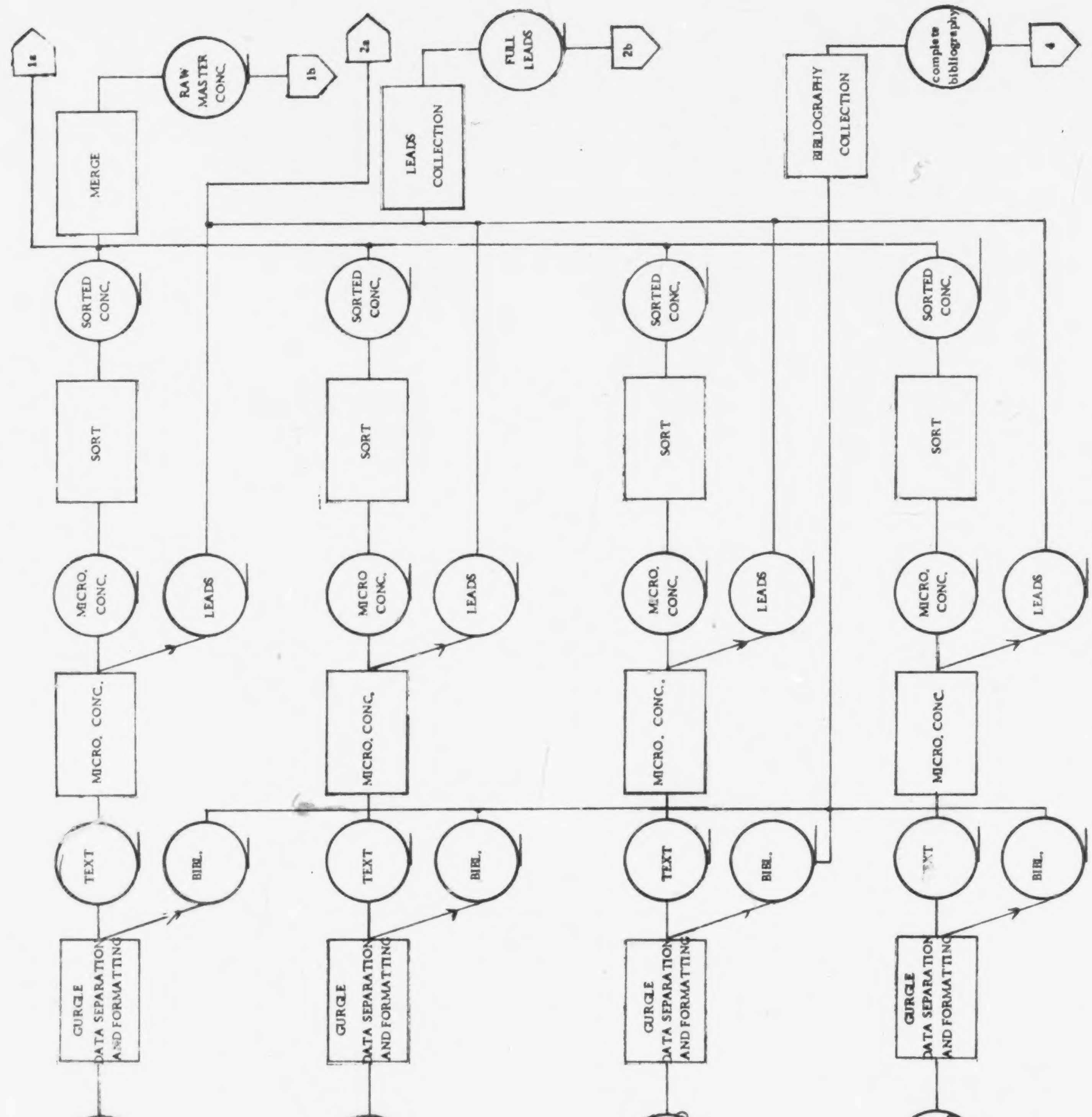

9tr IBM EL CODED
CODE CONV.
7tr HISI CODED
GURGLE DATA SEPARATION AND FORMATTING
TEXT
BIBL.
MICRO. CONC.
MICRO. CONC.
LEADS
SORT
SORTED CONC.
MERGE
RAW MASTER CONC.
1a
1b
2a
2b
LEADS COLLECTION
FULL LEADS
BIBLIOGRAPHY COLLECTION
complete bibliography
4
9tr IBM CODED

Figure V.1 (Continued).

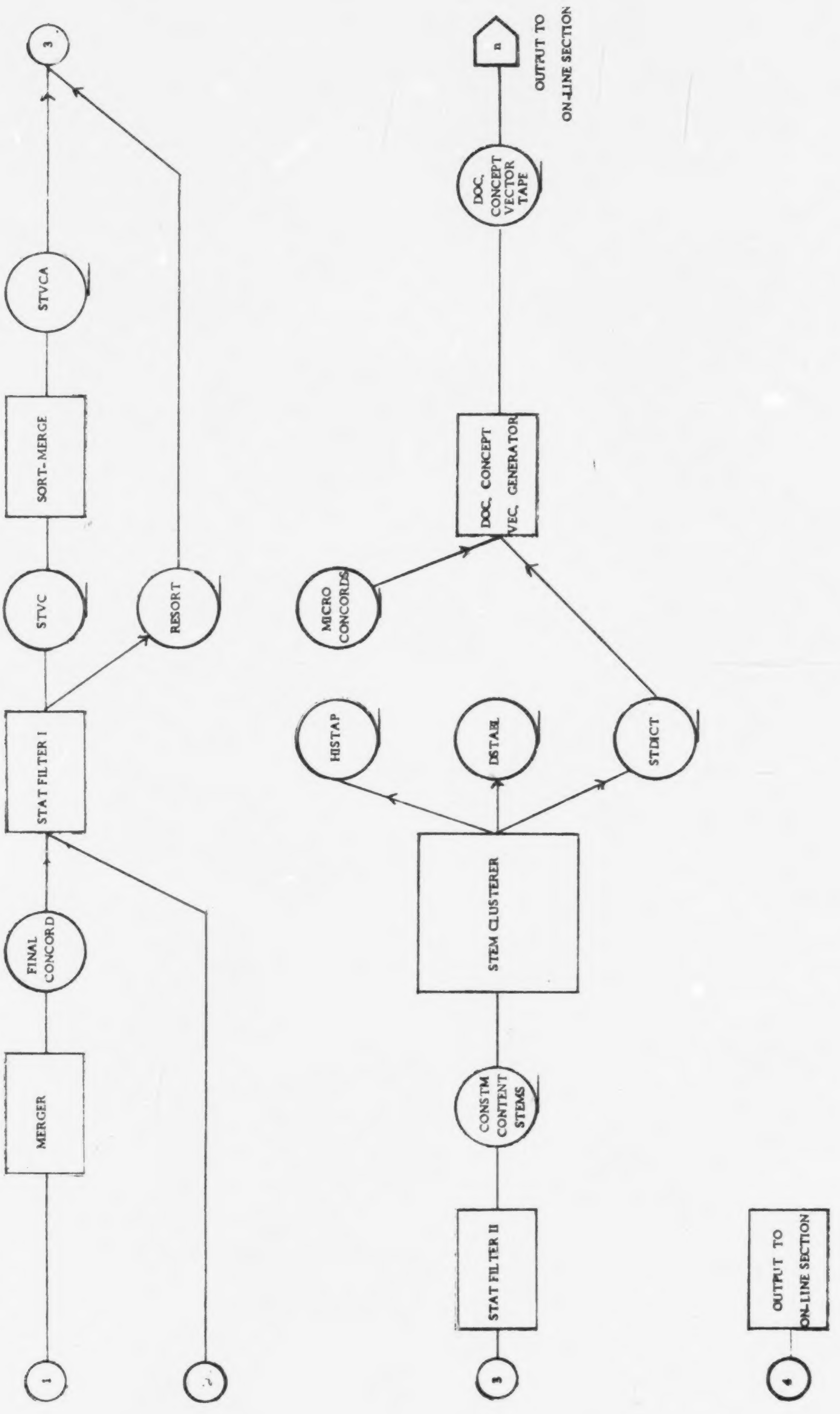

1
MERGER
FINAL CONCORD
STAT FILTER I
STVC
SORT-MERGE
STVCA
3
2
RESORT
3
STAT FILTER II
CONSTM CONTENT STEMS
STEM CLUSTERER
HISTAP
DSTABL
STDICT
MICRO CONCORDS
DOC. CONCEPT VEC. GENERATOR
DOC. CONCEPT VECTOR TAPE
n
OUTPUT TO ON-LINE SECTION
4
OUTPUT TO ON-LINE SECTION

Figure V.1. Completed.

subsequent data file has the same structure. (See Figure V.2 for a diagrammatic representation of a typical tape.)

During the conversion process all header and trailer labels are deleted from the text of the converted tape since they contain no pertinent data. The APL characters are converted to their equivalents in HISI six-bit code. The program description of CONVRT describes this process in more detail. The object of the conversion process is to obtain magnetic tape files structured in a compatible format, both to the HISI system and to the JOVIAL programs which have to process these data.

V.2.2 Flowchart and Description: CONVRT

This sub-section contains the flowchart for CONVRT, a detailed program description, and a summary of the production runs using this package, including a deck set-up. Figure V.3 is the flowchart, while Figure V.4 illustrates the production card deck structure.

V.2.2.1 Program Description of CONVRT. The following is a comprehensive discussion of the logic of the CONVRT package, keyed to the flowchart of Figure V.3. For the sake of efficiency, CONVRT was written in GMAP (HISI - 635 assembly language).

1. Write the dummy header label onto the output tape, file code 01. The 14 word label starts at location HEADER in the source code.
2. Write an EOF mark after the dummy header label.
3. Set item LABELS to 1. This variable stores the count of the number of labels which are to be read before a data file is encountered on the input tapes, which may have a multi-file structure. Each data file has a standard header and trailer label consisting of at least two physical records. It is therefore apparent that only one label must be read before the first data file is encountered and that two labels must be read before each subsequent data file is reached.
4. Set item LBLCNT, the counter which tallies the number of labels which have been read, to zero.

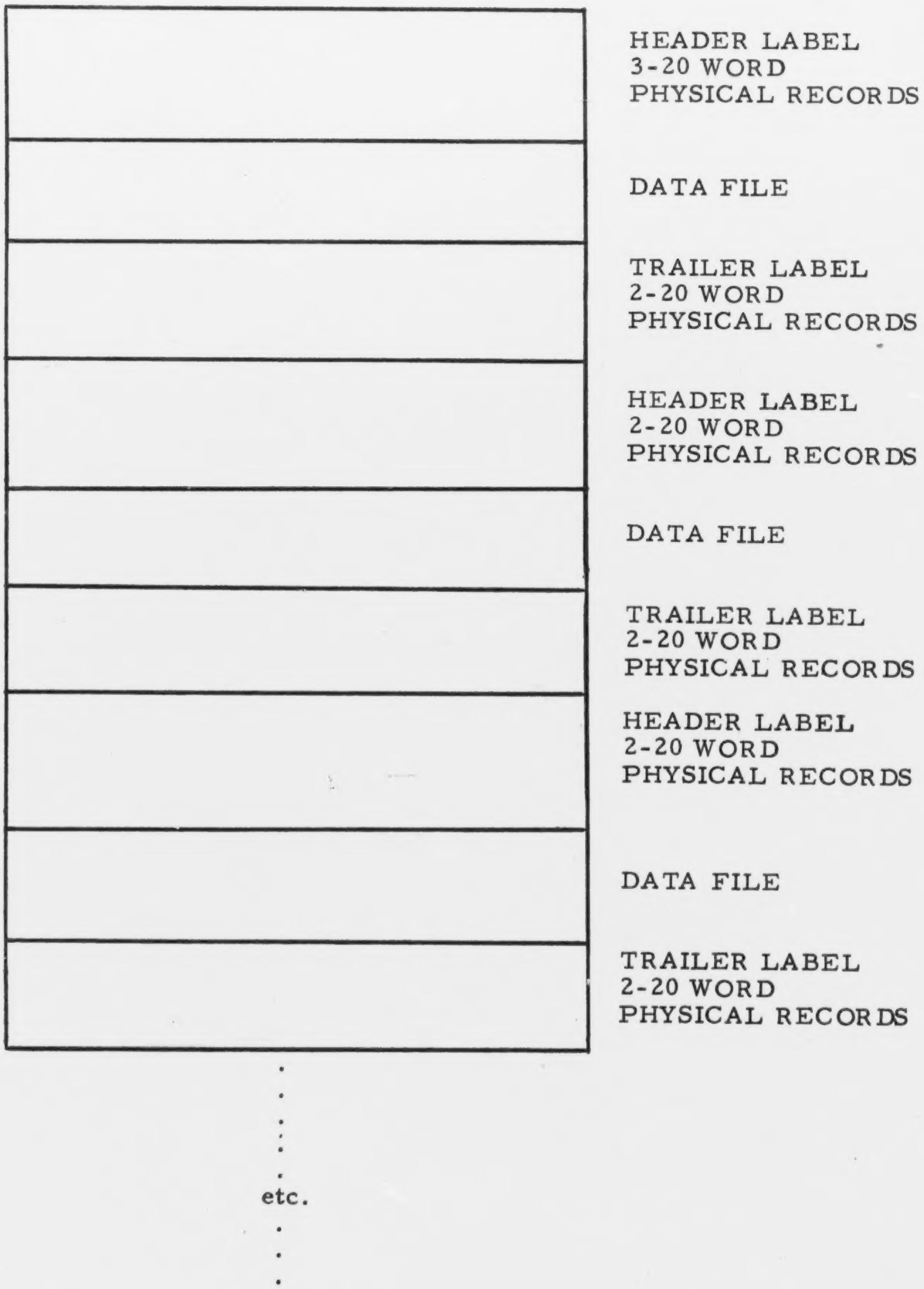

HEADER LABEL
3-20 WORD
PHYSICAL RECORDS
DATA FILE
TRAILER LABEL
2-20 WORD
PHYSICAL RECORDS
HEADER LABEL
2-20 WORD
PHYSICAL RECORDS
DATA FILE
TRAILER LABEL
2-20 WORD
PHYSICAL RECORDS
HEADER LABEL
2-20 WORD
PHYSICAL RECORDS
DATA FILE
TRAILER LABEL
2-20 WORD
PHYSICAL RECORDS
etc.

Figure V. 2. Structure of IBM 360, 9- Track Tapes

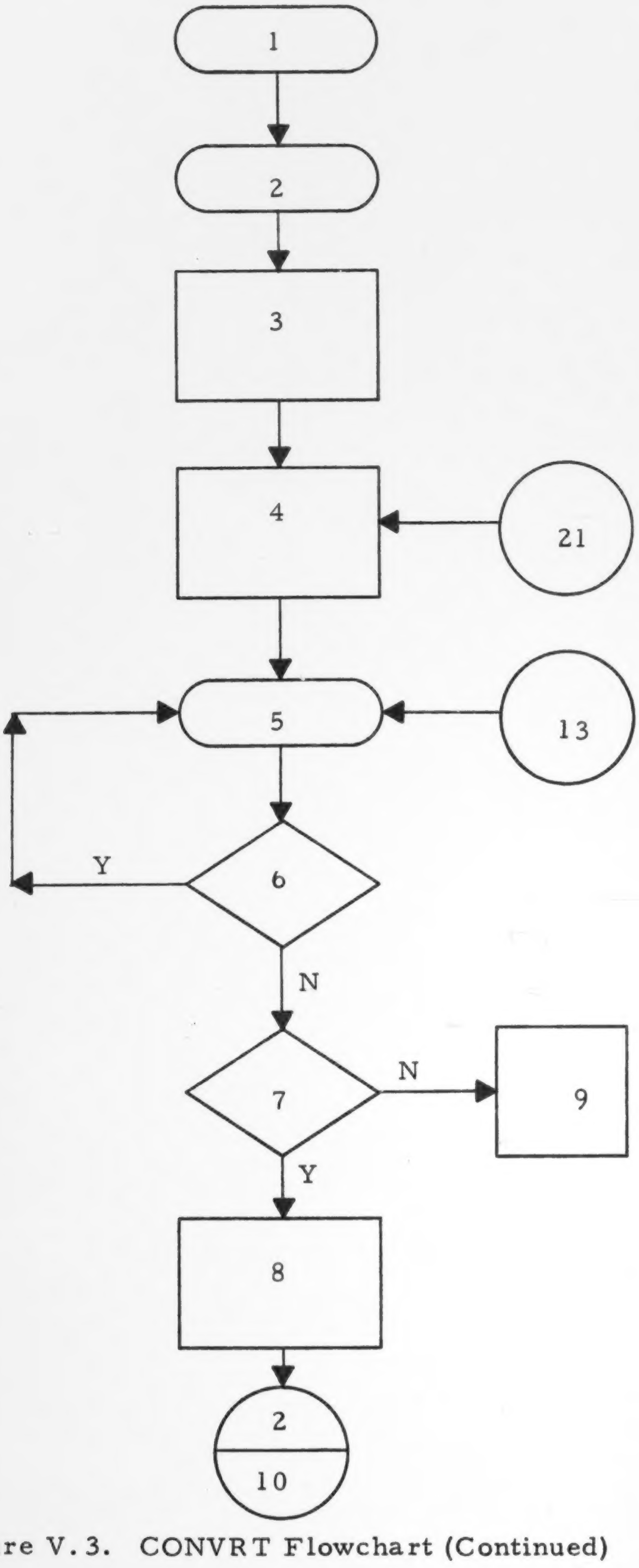

1
2
3
4
21
5
13
6
Y
N
7
N
9
Y
8
2
10

Figure V.3. CONVRT Flowchart (Continued)

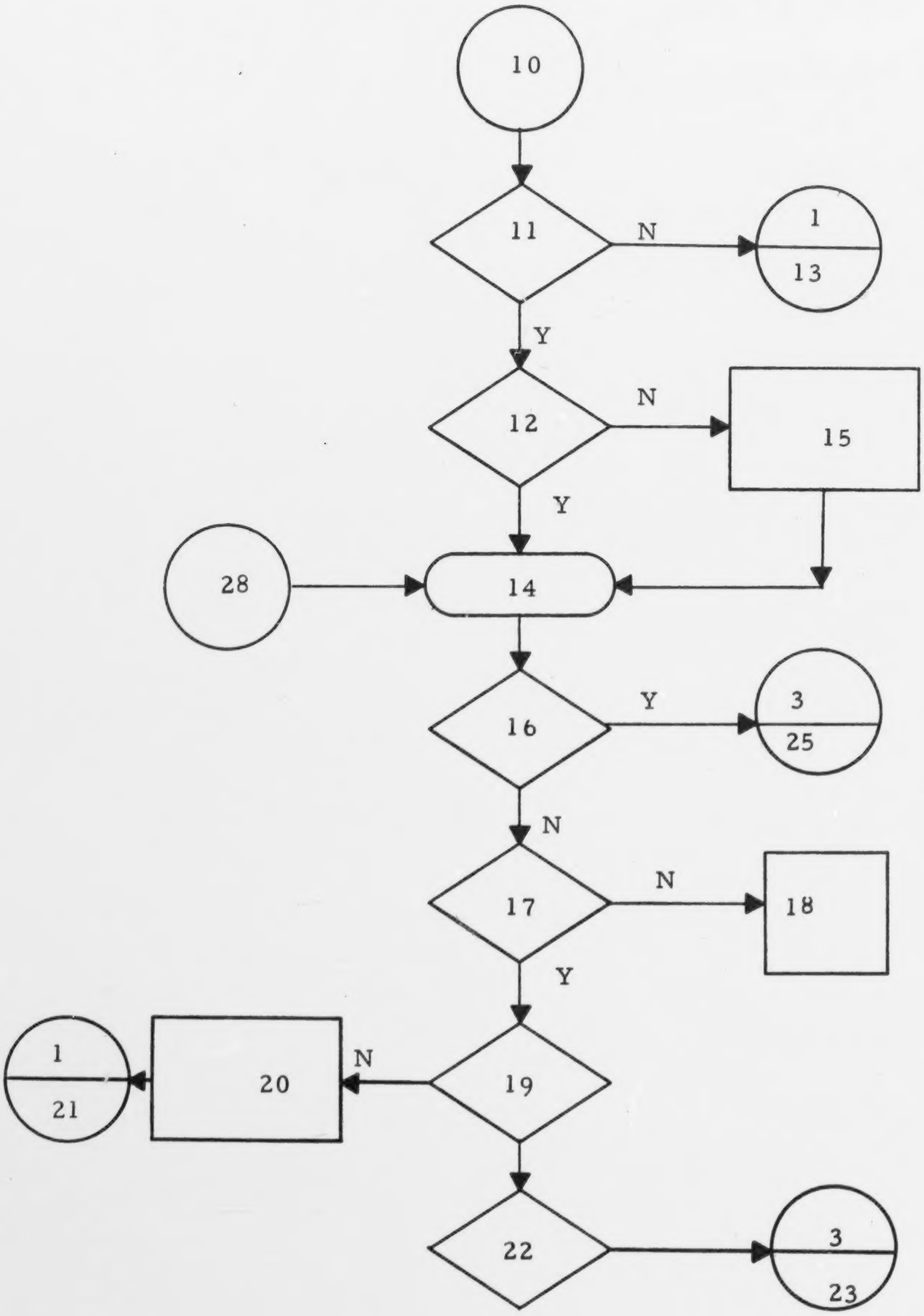

10
11
N
1
13
Y
12
N
15
Y
28
14
16
Y
3
25
N
17
N
18
Y
1
21
20
N
19
22
3
23

Figure V.3. CONVRT Flowchart (Continued)

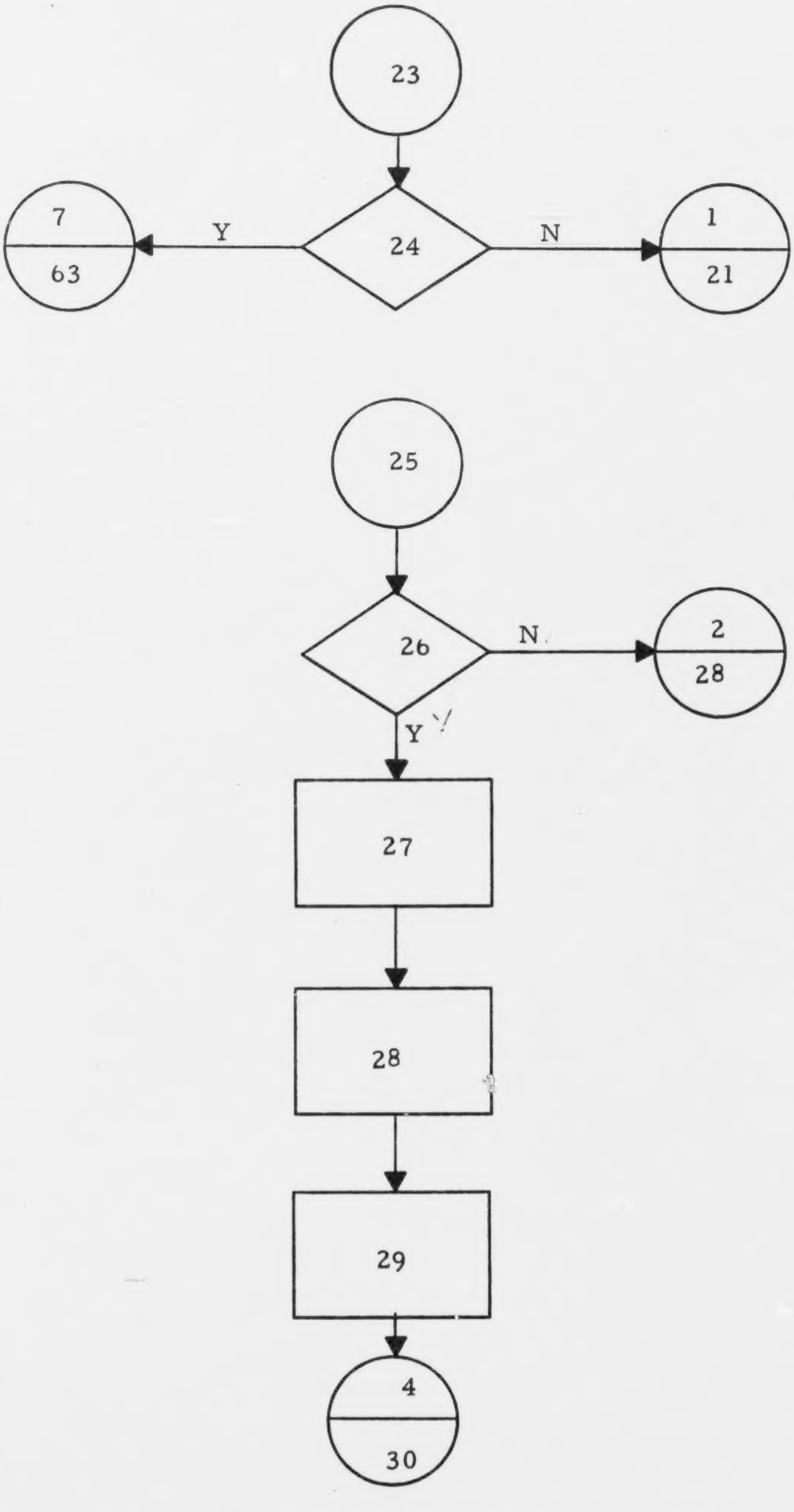

23
24
Y
N
7
63
1
21
25
26
N
2
28
Y
27
28
29
4
30

Figure V.3. CONVRT Flowchart (Continued)

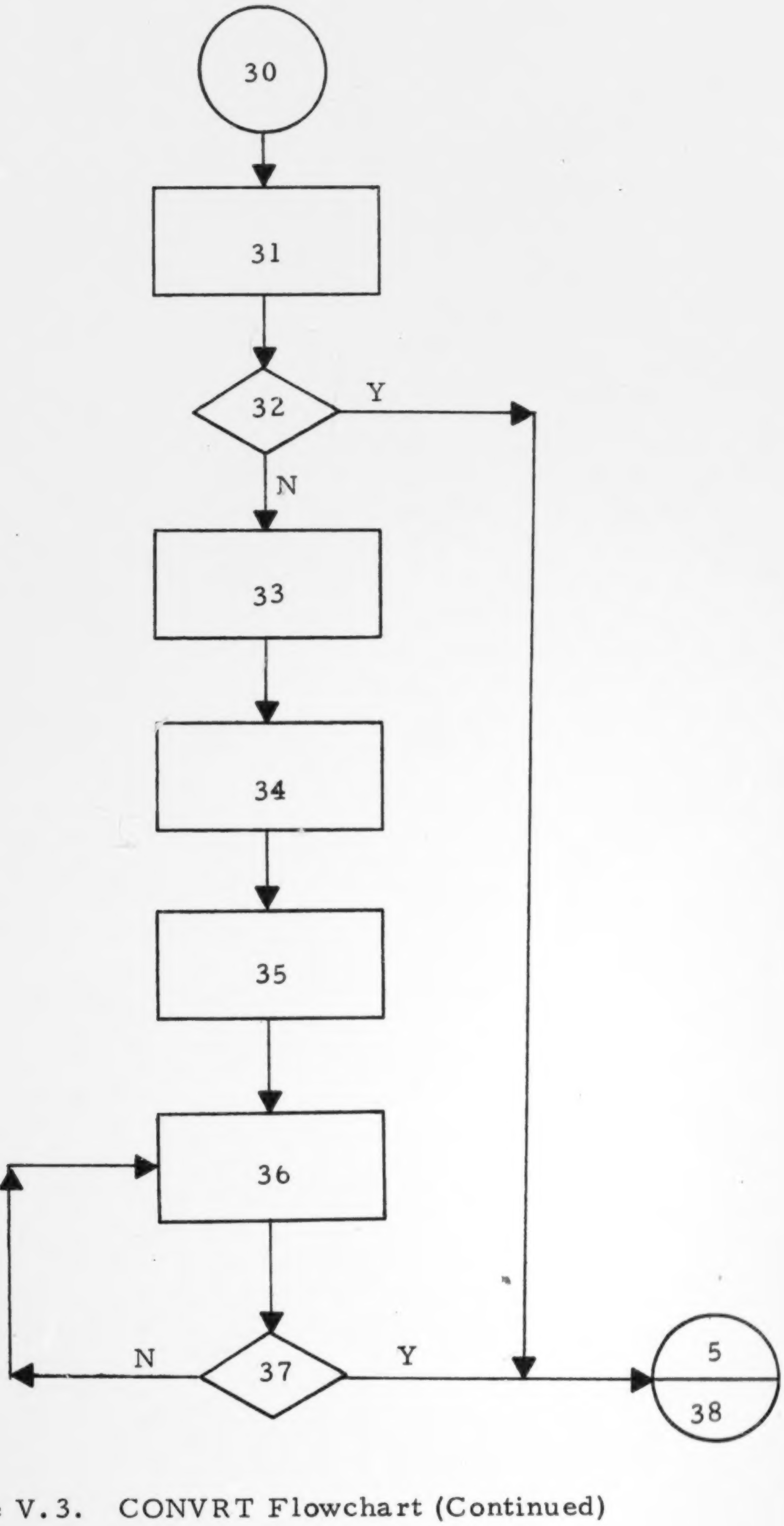

30
31
32
Y
N
33
34
35
36
N
37
Y
5
38

Figure V.3. CONVRT Flowchart (Continued)

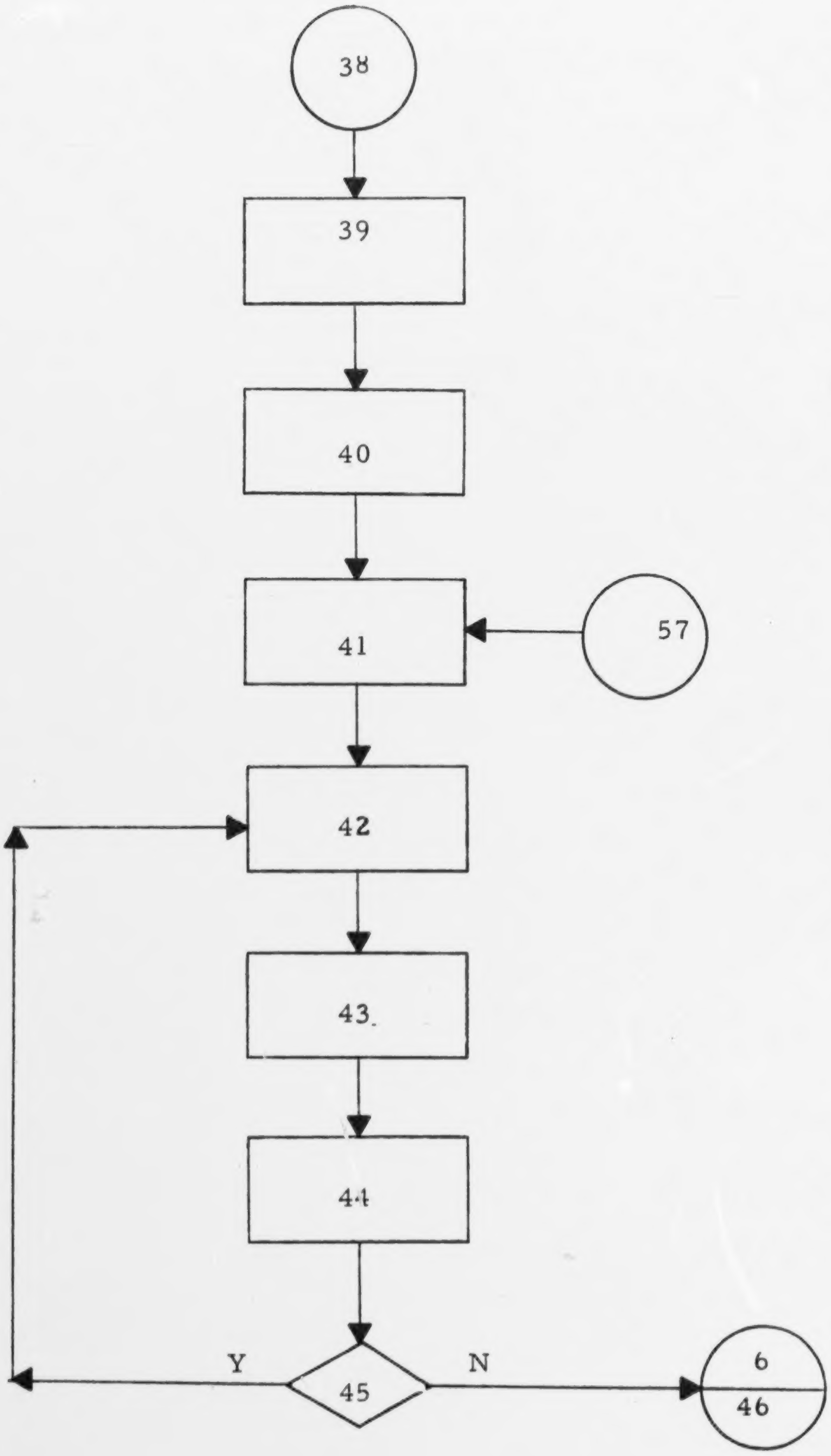

38
39
40
41
57
42
43
44
Y
45
N
6
46

Figure V.3. CONVRT Flowchart (Continued)

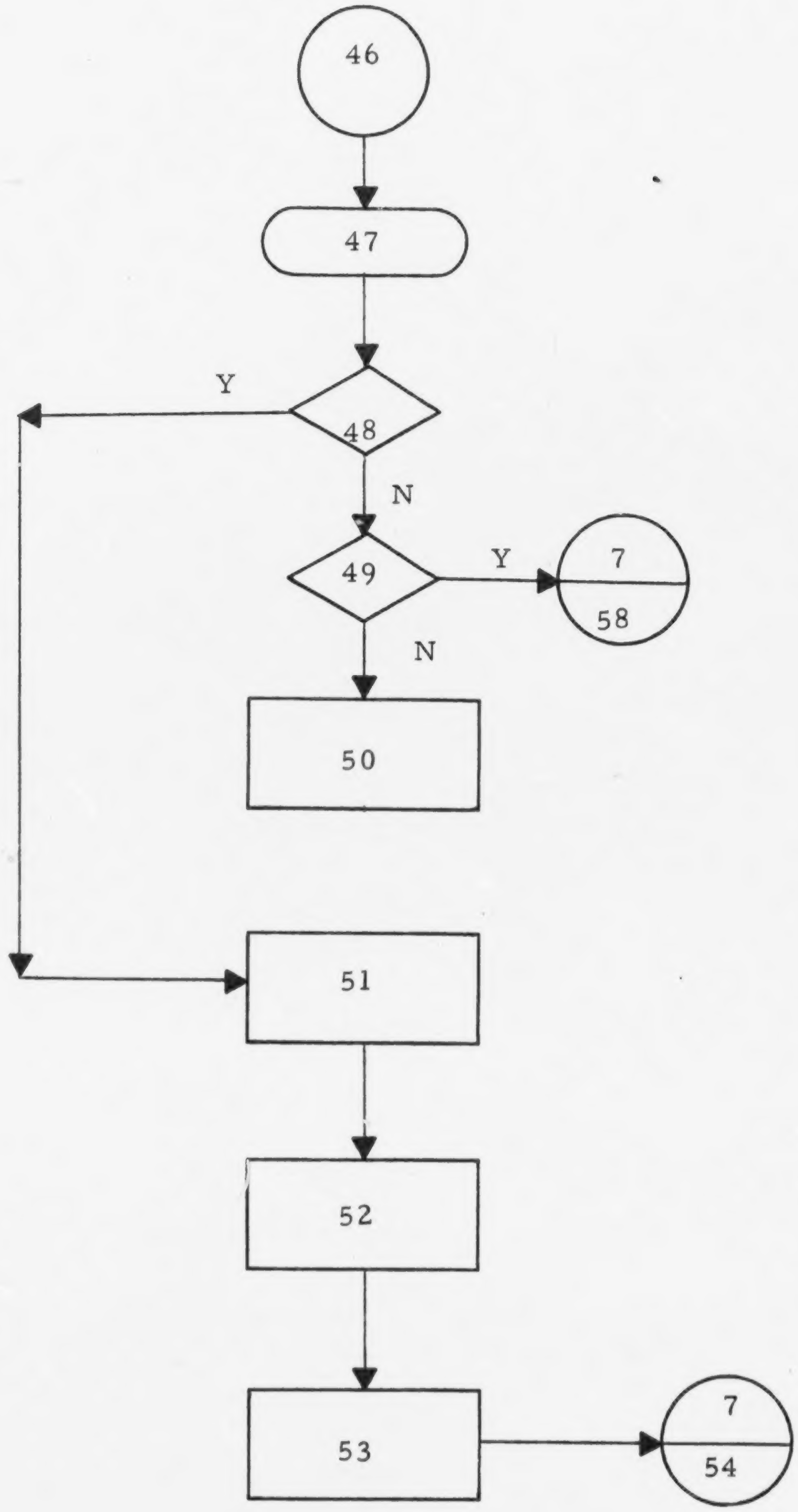

46
47
Y
48
N
49
Y
7
58
N
50
51
52
53
7
54

Figure V.3. CONVRT Flowchart (Continued)

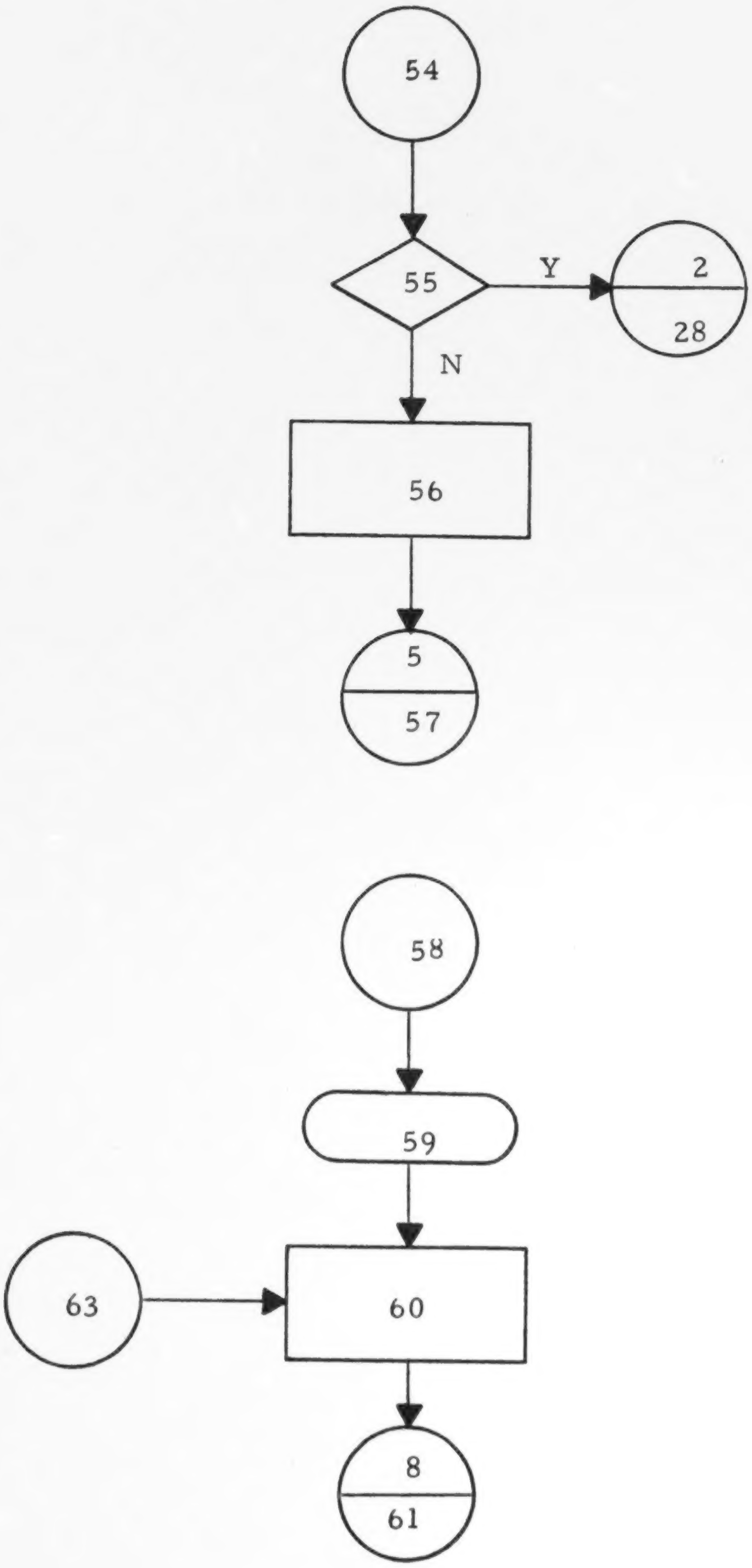

54
55
Y
2
28
N
56
5
57
58
59
63
60
8
61

Figure V.3. CONVRT Flowchart (Continued)

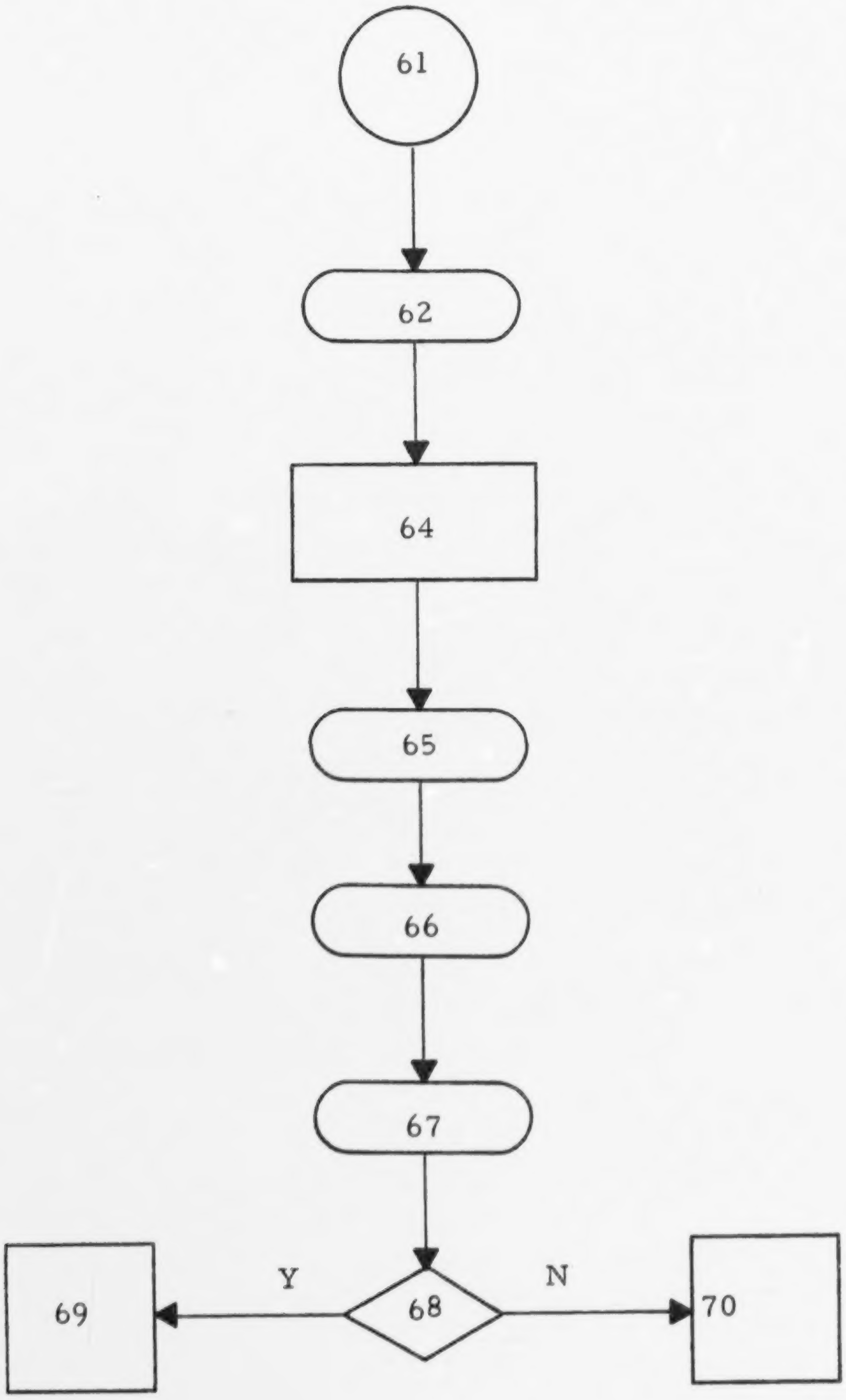
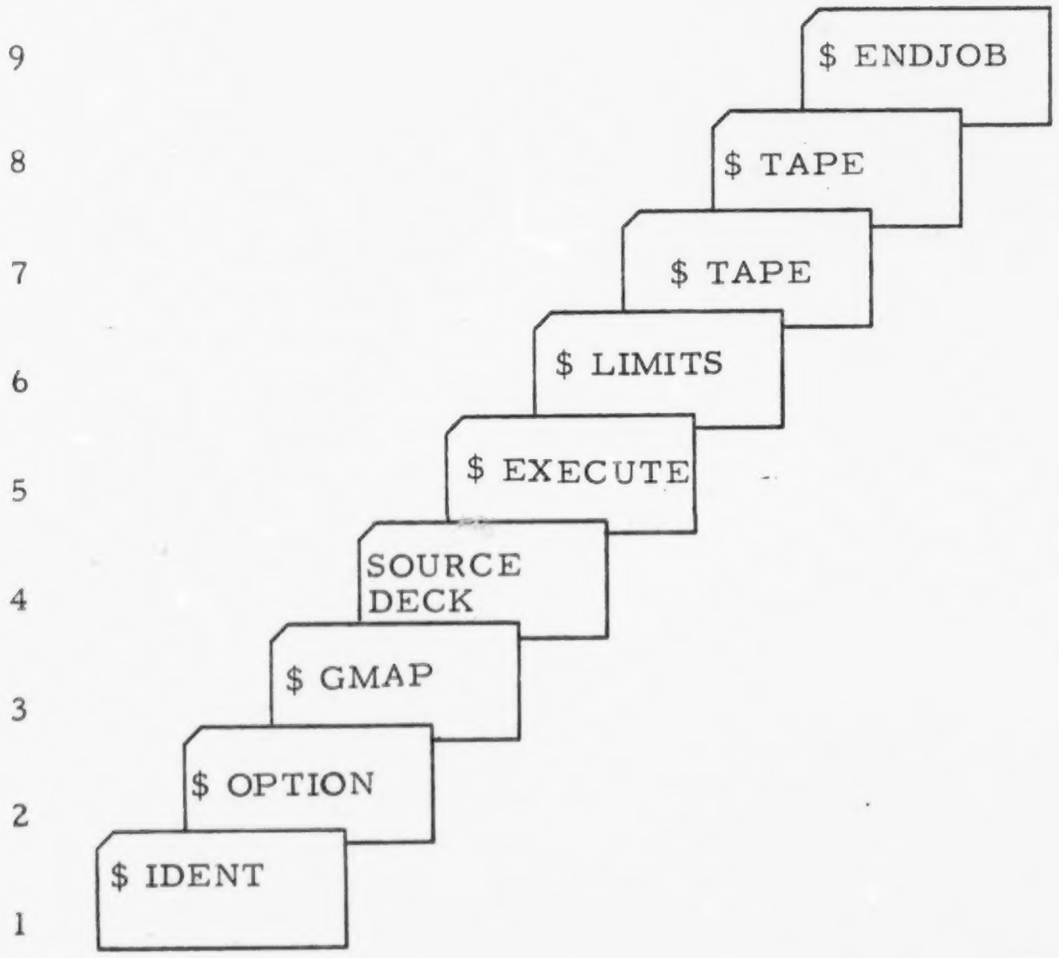

61
62
64
65
66
67
68
Y
N
69
70

Figure V.3. CONVRT Flowchart (Continued)


9
8
7
6
5
4
3
2
1
$ ENDJOB
$ TAPE
$ TAPE
$ LIMITS
$ EXECUTE
SOURCE DECK
$ GMAP
$ OPTION
$ IDENT

CARD NUMBER	FILE CODE	DESCRIPTION
7	01	Output Tape (7 Track)
8	X1	Input Tape (9 Track)

Figure V.4. Deck Structures for a CONVRT Run

5. Read a 20-word physical record from the label.

6. Test for an I/O MAJOR STATUS of "CHANNEL READY", which indicates that no I/O error or EOF mark has been detected. "CHANNEL READY"?

7. Test for a substatus of EOF, indicating end-of-label has been reached. If the substatus is not EOF, an I/O error has been detected. "EOF"? Yes => 8 No => 9

8. Increment LBLCNT by one.

9. An A4 MME GEBORT, indicating an I/O error during label manipulation, is executed.

11. Compare the number of labels which are to be read, stored in LABELS, with the number of labels which have been read, stored in LBLCNT. Have all labels been read? Yes =>12 No => 13

12. Test for LABELS = 2. If LABELS ≠ 2, it is set equal to two at this point since all following data files will be preceded by a trailer label of the previous data file and a header label. (See FCC 3).

14. Read a physical record of 8190 words.

15. Set LABELS = 2.

16. Check for a MAJOR STATUS OF "CHANNEL READY" (See FCC 6) "CHANNEL READY" Yes => 25 No =>17

17. Test for a substatus of EOF. EOF? Yes => 19 No => 18

18. An A2 MME GEBORT, indicating an I/O error during data input, is executed.

19. Test for FSKIP = 0. FSKIP is a constant set by the user, specifying the number of data files which are to be skipped. This procedure provides the user with a means of selecting specific files for transliteration. If no files are to be skipped, FSKIP is preset to zero. FSKIP = 0? Yes => 22 No => 20

20. Decrement FSKIP by one.

22. Increment the counter of files which have been read, item FILES.

24. Compare the number of files read, FILES, with the number of files specified to be read by the preset value stored in item FREAD. FREAD = FILES? Yes => 63 No =>21

26. Test for FSKIP = 0, indicating that all files to be skipped have been bypassed. FSKIP = 0? Yes => 27 No =>28

27. Preset the value of the number of TALLY passes to be made, LINT, and the count of the TALLY passes completed, INDEX, to zero.

28. Calculate the number of words which have been loaded into the input buffer by the last I/O call. This is done by subtracting the starting address of the input buffer, (IBM, OU), from the address of the last data word loaded into the buffer. (This value is stored in the second status return word of the I/O calling sequences). Store this value in parameter NWDS.

29. Calculate the number of characters loaded into the input buffer.

31. Calculate the number of TALLY passes necessary to transliterate the input buffer. The input buffer, which may contain up to 8190 words (32, 760 9-bit characters) of data is subdivided into 318 word modules (1, 908 6-bit characters) in order to conform to the GEFRC SSF physical record block size of 320 words (the first two words of the physical record are control words).

32. This test compares the number of words loaded into the input buffer, NWDS, with the number of words loaded by the previous I/O call, OLDWDS. If OLDWDS is greater than NWDS, then the overlapping data is filled with zeroes. NWDS $\geq$ OLDWDS? Yes => 38 No => 33

33. Store the number of overlapping words in parameter TIMES.

34. Construct the statement which performs the actual table clearing function. The mechanics of this procedure are as follows:

    A) Cell ZOUT1, which contains the instruction STZ IBM (STORE ZERO in the address IBM, the first cell of the input buffer), is loaded into the A-register.

    B) Bits 0-17 of the A-register now contain the starting address of the input buffer, IBM. To this address is added the value of the number of words transferred during the last I/O call, NWDS.

    C) The instruction now residing in the A-register is essentially STZ IBM+n, where "n" is the value stored in item NWDS. Address IBM+n is the location of the first overlapping cell in the input buffer.

D) The contents of the A-register are loaded into a cell labeled LOOP4 in the loop which executes the clearing procedure.

35. Preset item CHECK, the counter which tallies the reiterations of the table clearing loop, to zero.

36. Clear the cell in the buffer, increment the address field of the clearing instruction, and increment item CHECK.

37. Have all overlapping data words been cleared? No => 36 Yes => 38

39. Store the value of NWDS in OLDWDS.

40. Initialize the TALLYB (9-bit) pseudo-instruction. This instruction performs automatic character and address incrementation for 9-bit characters when used with the SC (sequential character) address modifier.

41. Initialize the TALLY (6-bit) pseudo-instruction. This instruction performs automatic character and address incrementation for 6-bit characters when used with the SC address modifier.

42. Load a 9-bit character from the input buffer.

43. Use the contents of the A-register as an address modifier in order to load the corresponding 6-bit character from the transliteration table CHARTB into the Q-register.

44. Store the contents of the Q-register, the transliterated character, in the output buffer.

45. Test for tally runout. If the tally runout indicator is off, the tally is not completed and transfer is made back to the initial operation of the loop. Tally runout indicator off? Yes => 42 No => 46

47. Write the output buffer of 320 words onto a magnetic tape.

48. Check for a MAJOR STATUS OF "CHANNEL READY" in the first status return word of the output calling sequence. "CHANNEL READY"? Yes => 51 No => 49

49. Check for a SUBSTATUS of EOT, indicating that the end-of-tape foil mask has been reached on the output tape. "EOT"? Yes => 59 No => 50

50. An A7 MME GEBORT is made which indicates that an I/O error has occurred during the output processing.

51. Increment the tally pass counter, INDEX.

52. Increment the physical record block serial number in the first simulated GEFRC control word of the output buffer.

53. Increment the physical record block count in the simulated GEFRC standard trailer label.

55. Test for complete transliteration of the input buffer. If LINT, the number of tally passes needed to transliterate the entire buffer, is equal to INDEX, the tally pass counter, transliteration is complete. LINT = INDEX? Yes => 28 No =>56

56. Reset the tally count field of the TALLYB pseudo-operation for 1908 characters (475 words).

59. Backspace two physical records in order to pass back over the physical end-of-tape foil marker on the output tape. Execution is terminated when the output tape is exhausted.

60. Generate the sentinel end-of-file record.

62. Write the sentinel EOF record onto the output tape.

64. Increment the physical record block count in the simulated GEFRC SSF trailer label.

65. Write an END-OF-FILE mark.

66. Write the simulated trailer label.

67. Write an END-OF-FILE MARK.

68. Test for exhausted output tape. Output tape exhausted? Yes => 69 No => 70

69. An ET MME GEBORT is made, indicating that the output tape reel was exhausted. When this occurs, the input tape must be reprocessed in contiguous segments.

70. An EF MME GEBORT is made, indicating that the entire input tape or that specified files of the input tape have been transliterated successfully.

V.2.2.2 Definitions of Data Descriptions Used in CONVRT.

CHARTB: The transliteration table. Each entry of this table is a right-justified BCD character which occupies a full 36-bit word. An 8-bit IBM character is used as an address modifier to fetch the corresponding 6-bit HISI character from the table (See FCC 43).

CHECK: A counter used during table clearing. As each word is cleared, CHECK is incremented. When CHECK is equal to TIMES, the number of words to be cleared, an exit is made from the loop (See FCC 35 and 36).

CONTRL: The two-word storage area which contains the simulated GEFRC output buffer control words and immediately proceeds the output buffer, GE. The first word, the physical record control word, has the following format:

bits 0-17	Physical record block serial number
bits 18-35	Block size in words (not including the control word)

The second word, the logical record control word, has the following format:

bits 0-17	Record size in words (not including the control word)
bits 24-29	Logical record type code
bits 30-35	Report code

(See FCC 52 and 60)

DCW1: The data control word specifying the location of the input buffer and the number of words to be read into the buffer (See FCC 14).

DCW2: The data control word specifying the location of the transliterated output buffer and the number of words to be transmitted (See FCC 47).

DCW3: The data control word specifying the location of the input buffer and the number of words to be read during the label skipping operation (See FCC 5).

DCW5: The data control word specifying the location of the standard trailer label and the number of words to be transmitted (See FCC 66).

DCW7: The data control word used to transmit the sentinel EOF physical record (See FCC 62).

DCW20: The data control word specifying the location of the standard header label and the number of words to be transmitted (See FCC 1).

FA: The cell containing the input file code.

FB: The cell containing the output file code.

FILES: The counter which tallies the transliterated files. The program is terminated when FILES is equal to FREAD, the number of files to be transliterated (See FCC 24).

FINI: A variable used as a counter when backspacing of the output file occurs. Two physical record blocks of 320 words each are backspaced in this operation; an exit is made from the loop when FINI = 2. When execution is to be terminated, FINI is tested in order to determine which exit abort code is to be transmitted.

FINI = 0. No backspacing of the output tape was done; the specified number of files to be transliterated have been successfully processed (EF ABORT CODE).

FINI ≠ 0. The output tape reel has been exhausted. The user must reprocess the input tape in segments (ET ABORT CODE).

See FCC 59, 68, 69, and 70)

FREAD: The number of files to be transliterated. This constant is preset by the user prior to running the program (See FCC 24).

FSKIP: The number of files to be bypassed. This constant is preset by the user prior to running the program (See FCC 19, 20, and 26).

GE: A 318 word storage area of the output buffer which immediately follows field CONTRL. The transliterated text is stored in the field GE.

HEADER: A 14 word storage area reserved for the simulated GEFRC SSF header label.

IBM: An 8192 word storage area reserved as the input buffer for the raw text from the 9 track IBM tape.

INC: A constant used to reset the TALLY COUNT of the TALLYB pseudo-operation. The tally count is segmented to correspond to the number of characters required to fill the output buffer. In order to resume processing, only the tally count must be reinitiated since the address field and the character field are preserved in the tally word (See FCC 56).

INDEX: A counter of the number of tally passes which have been made during the transliteration of a particular input buffer (See FCC 27, 51, and 55).

LABELS: The number of labels which must be read before a data file is reached on the input tape (See FCC 3, 11, 12, and 5).

LBLCNT: A counter which tallies the number of labels read when the tape is being advanced to the next data file (See FCC 4 and 11).

LINT: The number of TALLYB and TALLY passes required to transliterate the entire input buffer. The tally count field is preset for 1908 characters which is exactly the size of the 318 word data field of the output buffer (See FCC 27, 31, and 55).

NWDS: The number of words read into the input buffer (See FCC 28, 32, and 39).

OLDWDS: The number of words previously read into the previous input buffer (See FCC 32 and 39).

STAT1: The status return words for the I/O call which loads raw text from the input file into the input buffer (See FCC 14).

STAT2: The status return word for the I/O call which writes the contents of the output buffer onto magnetic tape (See FCC 47).

STAT3: The status return words for the I/O call which reads the labels of the input file during the label skipping operation (See FCC 5).

STAT4: The status return words for the I/O calls which write the end-of-file marks on the output tape after the header label, the data file, and the trailer label (See FCC 1, 2, 65, 66, and 67).

STAT5: The status return words for the I/O call which backspaces the output tape when an end-of-tape status is received. (See FCC 49 and 59).

STAT7: The status return words for the I/O call which transmits the sentinel EOF record (See FCC 62).

STAT8: The status return words for the I/O call which transmits the standard trailer label (See FCC 66).

STAT20: The status return words for the I/O call which writes the standard header label on the output tape (See FCC 1).

TALLY2: A dummy storage cell used to implement the TALLYB pseudo-instruction. The TALLYB instruction, stored in address TALLY3, is loaded into cell TALLY2 to initialize the tally function. TALLY2 contains the instruction which is actually operated upon during the tally procedure; TALLY3 is preserved for repeated usage (See FCC 40, 42, 45, and 56).

TALLY3: The cell containing the TALLYB pseudo-instruction which is used when 9 bit bytes are desired. The format of this instruction is as follows:

bits 0-17	The starting address of the tally operation for indirect reference
bits 18-29	The tally count field (the number of bytes to be referenced)
bits 33-35	Character position within the address specified in bits 0-17

When used with the SC (sequential character modifier), the HISI 635 performs the following functions automatically.

1) Add one to the character position field.
2) Subtract one from the tally count field.
3) Add one to the address field and set the character field to zero when a word boundary is crossed.

(See FCC 31, 40, 42, 45, and 56)

TALLY6: Same as TALLY2 but used with the TALLY pseudo-instruction for 6 bit bytes (See FCC 41).

TALLY5: The TALLY pseudo-instruction. The TALLY instruction operates in the same manner for 6 bit bytes as does the TALLYB function for 9 bit bytes (See FCC 41, 43, 44, and 45).

TIMES: The number of words to be cleared from the input buffer (See FCC 33 and 37).

TRAIL: The cell containing the EOF word used in building the end-of-file sentinel record (See FCC 60 and 62).

ZOUT1: The cell which contains the instruction used to initialize the input buffer clearing function (See FCC 34 and 36).

V.2.2.3 Preparation for a CONVRT Run. Prior to running program CONVRT the user must preset several items:

FSKIP	The number of data files to be skipped
FREAD	The number of files to be transliterated

These constants may be preset using the "OCT" or the "DEC" GMAP pseudo-instruction. Finally, words 3 and 4 of the dummy output file header label must contain the file serial number (tape reel number) which is entered in the TAPE control card. The file serial number is a BCD integer left-justified with blanks. (Example. The tape number 44321 would be coded as "OCT 200404030201".) No further set-up coding need be done aside from the $ TAPE control cards.

Routine CONVRT is an extremely efficient program which provides the user with a powerful tool for 9-TRACK to 7-TRACK tape conversions. Where different character sets are used all that is required is the modification of the transliteration table, CHARTB. Furthermore, the user is provided with a certain degree of flexibility in selecting specific files from his input tape for transliteration.

BLANK PAGE

V.2.2.4 Listing of Program CONVRT. This listing follows in Figure V.5:

```
10018 01  02-11-71   19,109

                                                      SYMDEF  OPEN
000000  000001 0010 00  000           OPEN     MME     GEINOS
000001  15 0000 000000  000                    WTB
000002  000246 000255   011                    ZERO    FB,DCW20
000003  000274 000000   010                    ZERO    STAT20
000004  000002 0010 00  000                    MME     GEROAD
000005  000001 0010 00  000                    MME     GEINOS
000006  55 0000 020001  000                    WEF
000007  000246 000000   010                    ZERO    FB
000010  000264 000000   010                    ZERO    STAT4
000011  000002 0010 00  000                    MME     GEROAD
000012  000001 2350 07  000                    LDA     1,DL
000013  000310 7550 00  010                    STA     LABELS          READ COUNT EQUAL TO ONE
000014  000311 4500 00  010           RESET    STZ     LBLCNT          INITIALIZE DUMMY READ COUNTER
000015  000001 0010 00  000           ADVNCE   MME     GEINOS          I/O REQUEST
000016  03 0000 000000  000                    RT9                     READ THE NINE-TRACK INPUT TAPE
000017  000245 000252   011                    ZERO    FA,DCW3
000020  000262 000000   010                    ZERO    STAT3
000021  000002 0010 00  000                    MME     GEROAD          ROADBLOCK
000022  022344 2360 00  010                    LDQ     =0607777777777  MASK ALL BITS BUT MAJOR STATUS
000023  000262 2350 00  010                    LDA     STAT3           LOAD THE STATUS WORD
000024  000000 2110 03  000                    CMK     0,DU            CHECK FOR MAJOR STATJS OF CHANNEL READY
000025  000015 6000 00  010                    TZE     ADVNCE          IF READY, READ NEXT RECORD OF THE LABEL
000026  040000 2110 03  000                    CMK     =0040000,DU     CHECK FOR END-OF-FILE
000027  000040 6010 00  010                    TNZ     ABORT4          IF NOT EOF, TRANSFER TO USER ABORT A4
000030  000311 0540 00  010                    AOS     LBLCNT          ANOTHER LABEL HAS BEEN READ
000031  000310 2350 00  010                    LDA     LABELS          LOAD COUNT OF NO, OF LABELS TO BE READ
000032  000311 1150 00  010                    CMPA    LBLCNT          COMPARE WITH LABELS ALREADY READ
000033  000015 6010 00  010                    TNZ     ADVNCE          ANOTHER LABEL MUST BE READ
000034  000310 2350 00  010                    LDA     LABELS          IF THE LABELS TO BE READ IS 2, THEN
000035  000002 1150 07  000                    CMPA    2,DL            SOME INITIALIZATION STEPS ARE SKIPPED
000036  000042 6010 00  010                    TNZ     START1          TRANSFER TO INITIALIZATION INSTRUCTIONS
000037  000044 7100 00  010                    TRA     BEGIN           SKIP THE INITIALIZATION INSTRUCTIONS
000040  202104 2360 07  000           ABORT4   LDQ     =3H A4,DL       SET THE ABORT CODE
000041  000010 0010 00  000                    MME     GEBORT          ABORT
000042  000002 2350 07  000           START1   LDA     2,DL            SET LABELS TO BE DUMMY READ TO 2 IN
000043  000310 7550 00  010                    STA     LABELS          ORDER TO BYPASS INTER-FILE LABELS
000044  000001 0010 00  000           BEGIN    MME     GEINOS          MASTER MODE ENTRY I/O REQUEST
000045  03 0000 000000  000                    RT9                     READ THE 9 TRACK INPJT TAPE
000046  000245 000247   011                    ZERO    FA,DCW1
000047  000256 000000   010                    ZERO    STAT1
000050  000002 0010 00  000                    MME     GEROAD          ROADBLOCK
000051  022344 2360 00  010                    LDQ     =0607777777777  MASK ALL BITS BUT MAJOR STATUS
000052  000256 2350 00  010                    LDA     STAT1           LOAD THE STATUS WORD
000053  000000 2110 03  000                    CMK     0,DU            CHECK FOR STATUS OF CHANNEL READY
000054  000073 6000 00  010                    TZE     CONT            TRANSFER TO PROCESS INPUT BUFFER
000055  040000 2110 03  000                    CMK     =0040000,DU     CHECK FOR EOF STATUS
000056  000071 6010 00  010                    TNZ     ABORT3          TRANSFER  IF NOT EOF OR READY STATUS
000057  022340 2350 00  010                    LDA     FSKIP
000060  000064 6000 00  010                    TZE     SKIPED
000061  000001 1750 07  000                    SBA     1,DL
000062  022340 7550 00  010                    STA     FSKIP
```

Figure V.5. CONVRT Listing

```
10018 01  02-11-71   19,109

    000063  000014 7100 00  010     53         TRA    RESET
    000064  000313 0540 00  010     54 SKIPED  AOS    FILES
    000065  022341 2350 00  010     55         LDA    FREAD          LOAD PRESET VALUE OF DATA FILES TO READ
    000066  000313 1150 00  010     56         CMPA   FILES          COMPARE WITH NUMBER OF FILES READ
    000067  000202 6000 00  010     57         TZE    IEOF           IF ALL FILES ARE READ, TERMINATE THE JOB
    000070  000014 7100 00  010     58         TRA    RESET          TRANSFER TO READ LABELS
    000071  202102 2360 07  000     59 ABORT3  LDQ    =3H A2,DL
    000072  000010 0010 00  000     60         MME    GEBORT
    000073  022340 2350 00  010     61 CONT    LDA    FSKIP
    000074  000044 6010 00  010     62         TNZ    BEGIN
    000075  000304 4500 00  010     63         STZ    LINT
    000076  000305 4500 00  010     64         STZ    INDEX          INITIALIZE TALLY PASS COUNTER
    000077  000257 2220 00  010     65         LDX2   STAT1+1        LOAD SECOND  INPUT STATUS WORD
    000100  000720 1620 03  010     66         SBX2   IBM,DU         SUBTRACT INITIAL ADDRESS OF INPUT BUFFER
    000101  000303 4420 00  010     67         SXL2   NWDS           STORE COUNT OF WORDS IN BUFFER
    000102  000303 2360 00  010     68         LDQ    NWDS           LOAD COUNT OF WORDS IN THE INPUT BUFFER
    000103  000002 7360 00  000     69         QLS    2              MULTIPLY BY FOUR FOR CHARACTER COUNT
    000104  022345 5060 00  010     70         DIV    =1908          CALC TALLY BOUNDS FOR THIS BUFFER
    000105  000304 7560 00  010     71         STQ    LINT           STORE COUNT OF STANDARD TALLY PASSES
    000106  000304 0540 00  010     72         AOS    LINT
    000107  000303 2350 00  010     73         LDA    NWDS           THE FOLLOWING CODE ZEROES THE INPUT AND
    000110  000307 1750 00  010     74         SBA    OLDWDS         OUTPUT BUFFERS IF THE PREVIOUS RECORD
    000111  000130 6050 00  010     75         TPL    PASS           CONTAINED MORE WORDS THAN THE CURRENT
    000112  000000 5310 00  000     76         NEG                   RECORD,
    000113  000306 7550 00  010     77         STA    TIMES
    000114  000303 2350 00  010     78         LDA    NWDS
    000115  000022 7350 00  000     79         ALS    18
    000116  000317 0750 00  010     80         ADA    ZOUT1
    000117  000121 7550 00  010     81         STA    LOOP4
    000120  000314 4500 00  010     82         STZ    CHECK
    000121  000000 000000   000     83 LOOP4   ZERO
    000122  000315 2350 00  010     84         LDA    ONE
    000123  000121 0550 00  010     85         ASA    LOOP4
    000124  000314 0540 00  010     86         AOS    CHECK
    000125  000314 2350 00  010     87         LDA    CHECK
    000126  000306 1750 00  010     88         SBA    TIMES
    000127  000121 6040 00  010     89         TMI    LOOP4
    000130  000303 2350 00  010     90 PASS    LDA    NWDS           RESET THE OLD WORD COUNT AND CONTINUE
    000131  000307 7550 00  010     91         STA    OLDWDS         ON TO THE ACTUAL CONVERSION AREA
    000132  000300 2350 00  010     92         LDA    TALLY3         LOAD TALLYB WORD (9-BIT)
    000133  000277 7550 00  010     93         STA    TALLY2         STORE IN TALLY2
    000134  000301 2350 00  010     94 LOOP0   LDA    TALLY5         LOAD TALLY (6-BIT)
    000135  000302 7550 00  010     95         STA    TALLY6         STORE OUTPUT BUFFER TALLY
    000136  000277 2350 52  010     96 LOOP1  ELDA    TALLY2,SC      LOAD 9-BIT CHAR FROM INPUT BUFFER
    000137  000320 2360 05  010     97         LDQ    CHARTB,AL      LOAD WITH CONVERTED CHARACTER FROM TABLE
    000140  000302 7560 52  010     98         STQ    TALLY6,SC      STORE IN OUTPUT BUFFER
    000141  000136 6070 00  010     99         TTF    LOOP1          REPEAT UNTIL TALLY RJNOUT
    000142  000001 0010 00  000    100         MME    GEINOS
    000143  15 0000 000000  000    101         WTB                   WRITE A BINARY TAPE
    000144  000246 000251   011    102         ZERO   FB,DCW2
    000145  000260 000000   010    103         ZERO   STAT2
    000146  000002 0010 00  000    104         MME    GEROAD         ROADBLOCK
```

Figure V.5.   (Continued)

```
10018 01  02-11-71   19,109

     000147  022344 2360 00  010    105          LDQ     =0607777777777
     000150  000260 2350 00  010    106          LDA     STAT2
     000151  000000 2110 03  000    107          CMK     0,DU
     000152  000157 6000 00  010    108          TZE     AGAIN
     000153  030000 2110 03  000    109          CMK     =0030000,DU
     000154  000171 6000 00  010    110          TZE     EOT
     000155  202107 2360 07  000    111          LDQ     =3H A7,DL
     000156  000010 0010 00  000    112          MME     GEBORT
     000157  000305 0540 00  010    113 AGAIN    AOS     INDEX               INCREMENT TALLY PASS COUNTER
     000160  000315 2350 00  010    114          LDA     ONE                 INCREMENT SIMULATED
     000161  021620 0550 00  010    115          ASA     CONTRL              GEFRC RECORD COUNT
     000162  022337 0540 00  010    116          AOS     TRAIL+1             INCREMENT SIMULATED 3LOCK COUNT
     000163  000304 2350 00  010    117          LDA     LINT                COMPARE WITH STANDARD PASSES REQUIRED
     000164  000305 1750 00  010    118          SBA     INDEX               THE NUMBER OF PASSES MADE
     000165  000044 6000 00  010    119          TZE     BEGIN
     000166  000316 2350 00  010    120          LDA     INC                 LOAD NORMAL TALLY COJNTER
     000167  000277 0550 00  010    121          ASA     TALLY2              RESET TALLY COUNTER JF TALLY2 WORD
     000170  000134 7100 00  010    122          TRA     LOOP0               RESUME CHARACTER CONVERSION
     000171  000001 0010 00  000    123 EOT      MME     GEINOS
     000172  46 0000 020001  000    124          BSR
     000173  000246 000000   010    125          ZERO    FB
     000174  000266 000000   010    126          ZERO    STAT5
     000175  000002 0010 00  000    127          MME     GERDAD
     000176  000312 0540 00  010    128          AOS     FINI
     000177  000002 2350 07  000    129          LDA     2,DL
     000200  000312 1150 00  010    130          CMPA    FINI
     000201  000171 6010 00  010    131          TNZ     EOT
     000202  022346 2350 00  010    132 IEOF     LDA     =0202546262020
     000203  021622 7550 00  010    133          STA     QE
     000204  021620 2350 00  010    134          LDA     CONTRL
     000205  022347 1750 00  010    135          SBA     =0000000080475
     000206  021620 7550 00  010    136          STA     CONTRL
     000207  021621 2350 00  010    137          LDA     CONTRL+1
     000210  022350 1750 00  010    138          SBA     =0000475080000
     000211  021621 7550 00  010    139          STA     CONTRL+1
     000212  000001 0010 00  000    140          MME     GEINOS
     000213  15 0000 000000  000    141          WTB
     000214  000246 000254   011    142          ZERO    FB,DCW7
     000215  000270 000000   010    143          ZERO    STAT7
     000216  000002 0010 00  000    144          MME     GERDAD
     000217  022337 0540 00  010    145          AOS     TRAIL+1
     000220  000001 0010 00  000    146          MME     GEINOS
     000221  55 0000 020001  000    147          WEF
     000222  000246 000000   010    148          ZERO    FB
     000223  000264 000000   010    149          ZERO    STAT4
     000224  000002 0010 00  000    150          MME     GERDAD
     000225  000001 0010 00  000    151          MME     GEINOS
     000226  15 0000 000000  000    152          WTB
     000227  000246 000253   011    153          ZERO    FB,DCW5
     000230  000272 000000   010    154          ZERO    STAT8
     000231  000002 0010 00  000    155          MME     GERDAD
     000232  000001 0010 00  000    156          MME     GEINOS
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,109

000233  55 0000 020001  000   157         WEF
000234  000246 000000   010   158         ZERO    FB
000235  000264 000000   010   159         ZERO    STAT4
000236  000002 0010 00  000   160         MME     GEROAD
000237  000312 2340 00  010   161         SZN     FINI
000240  000243 6010 00  010   162         TNZ     ALTER
000241  202526 2360 07  000   163         LDQ     =3H EF,DL
000242  000010 0010 00  000   164         MME     GEBORT
000243  202563 2360 07  000   165 ALTER   LDQ     =3H ET,DL
000244  000010 0010 00  000   166         MME     GEBORT
000245  000000006701    000   167 FA      BCI     1,0000X1
000246  000000000001    000   168 FB      BCI     1,000001
000247  000720 010000   010   169 DCW1    IOTP    IBM,4096
000250  010720 000000   010   170         IOTD    IBM+4096,4096
000251  021620 000500   010   171 DCW2    IOTD    CONTRL,320
000252  000720 000024   010   172 DCW3    IOTD    IBM,20
000253  022336 000002   010   173 DCW5    IOTD    TRAIL,2
000254  021620 000003   010   174 DCW7    IOTD    CONTRL,3
000255  022320 000016   010   175 DCW20   IOTD    HEADER,14
                000256        176 STAT1   BSS     2
                000260        177 STAT2   BSS     2
                000262        178 STAT3   BSS     2
                000264        179 STAT4   BSS     2
                000266        180 STAT5   BSS     2
                000270        181 STAT7   BSS     2
                000272        182 STAT8   BSS     2
                000274        183 STAT20  BSS     2
000276  272520200600    000   184 LABEL   OCT     272520200600
000277  000000 000000   000   185 TALLY2  ZERO
000300  000720 3564 40  010   186 TALLY3  TALLYB  IBM,1908,0
000301  021622 3564 00  010   187 TALLY5  TALLY   GE,1908,0
000302  000000 000000   000   188 TALLY6  ZERO
                000303        189 NWDS    BSS     1
                000304        190 LINT    BSS     1
                000305        191 INDEX   BSS     1
                000306        192 TIMES   BSS     1
                000307        193 OLDWDS  BSS     1
                000310        194 LABELS  BSS     1
                000311        195 LBLCNT  BSS     1
000312  000000000000    000   196 FINI    OCT     0
000313  000000000000    000   197 FILES   OCT     0
000314  000000000000    000   198 CHECK   OCT     0
000315  000001000000    000   199 ONE     OCT     000001000000
000316  000000356400    000   200 INC     OCT     000000356400
000317  000720 4500 00  010   201 ZOUT1   STZ     IBM
000320  000000000020    000   202 CHARTB  OCT     20
000321  000000000020    000   203         OCT     20
000322  000000000020    000   204         OCT     20
000323  000000000020    000   205         OCT     20
000324  000000000020    000   206         OCT     20
000325  000000000020    000   207         OCT     20
000326  000000000020    000   208         OCT     20
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,109

        000327  000000000020    000  209        OCT     20
        000330  000000000020    000  210        OCT     20
        000331  000000000020    000  211        OCT     20
        000332  000000000020    000  212        OCT     20
        000333  000000000020    000  213        OCT     20
        000334  000000000020    000  214        OCT     20
        000335  000000000020    000  215        OCT     20
        000336  000000000020    000  216        OCT     20
        000337  000000000020    000  217        OCT     20
        000340  000000000020    000  218        OCT     20
        000341  000000000020    000  219        OCT     20
        000342  000000000020    000  220        OCT     20
        000343  000000000020    000  221        OCT     20
        000344  000000000020    000  222        OCT     20
        000345  000000000020    000  223        OCT     20
        000346  000000000020    000  224        OCT     20
        000347  000000000020    000  225        OCT     20
        000350  000000000020    000  226        OCT     20
        000351  000000000020    000  227        OCT     20
        000352  000000000020    000  228        OCT     20
        000353  000000000020    000  229        OCT     20
        000354  000000000020    000  230        OCT     20
        000355  000000000020    000  231        OCT     20
        000356  000000000020    000  232        OCT     20
        000357  000000000020    000  233        OCT     20
        000360  000000000020    000  234        OCT     20
        000361  000000000020    000  235        OCT     20
        000362  000000000020    000  236        OCT     20
        000363  000000000020    000  237        OCT     20
        000364  000000000020    000  238        OCT     20
        000365  000000000020    000  239        OCT     20
        000366  000000000020    000  240        OCT     20
        000367  000000000020    000  241        OCT     20
        000370  000000000020    000  242        OCT     20
        000371  000000000020    000  243        OCT     20
        000372  000000000020    000  244        OCT     20
        000373  000000000020    000  245        OCT     20
        000374  000000000020    000  246        OCT     20
        000375  000000000020    000  247        OCT     20
        000376  000000000020    000  248        OCT     20
        000377  000000000020    000  249        OCT     20
        000400  000000000020    000  250        OCT     20
        000401  000000000020    000  251        OCT     20
        000402  000000000020    000  252        OCT     20
        000403  000000000020    000  253        OCT     20
        000404  000000000020    000  254        OCT     20
        000405  000000000020    000  255        OCT     20
        000406  000000000020    000  256        OCT     20
        000407  000000000020    000  257        OCT     20
        000410  000000000020    000  258        OCT     20
        000411  000000000020    000  259        OCT     20
        000412  000000000020    000  260        OCT     20
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,109

      000413  000000000020    000   261        OCT    20
      000414  000000000020    000   262        OCT    20
      000415  000000000020    000   263        OCT    20
      000416  000000000020    000   264        OCT    20
      000417  000000000020    000   265        OCT    20
      000420  000000000020    000   266        OCT    20
      000421  000000000020    000   267        OCT    20
      000422  000000000020    000   268        OCT    20
      000423  000000000020    000   269        OCT    20
      000424  000000000020    000   270        OCT    20
      000425  000000000020    000   271        OCT    20
      000426  000000000020    000   272        OCT    20
      000427  000000000020    000   273        OCT    20
      000430  000000000020    000   274        OCT    20
      000431  000000000020    000   275        OCT    20
      000432  000000000020    000   276        OCT    20
      000433  000000000033    000   277        OCT    33
      000434  000000000020    000   278        OCT    20
      000435  000000000035    000   279        OCT    35
      000436  000000000060    000   280        OCT    60
      000437  000000000020    000   281        OCT    20
      000440  000000000020    000   282        OCT    20
      000441  000000000020    000   283        OCT    20
      000442  000000000020    000   284        OCT    20
      000443  000000000020    000   285        OCT    20
      000444  000000000020    000   286        OCT    20
      000445  000000000020    000   287        OCT    20
      000446  000000000020    000   288        OCT    20
      000447  000000000020    000   289        OCT    20
      000450  000000000020    000   290        OCT    20
      000451  000000000020    000   291        OCT    20
      000452  000000000020    000   292        OCT    20
      000453  000000000053    000   293        OCT    53
      000454  000000000054    000   294        OCT    54
      000455  000000000055    000   295        OCT    55
      000456  000000000020    000   296        OCT    20
      000457  000000000020    000   297        OCT    20
      000460  000000000052    000   298        OCT    52
      000461  000000000061    000   299        OCT    61
      000462  000000000020    000   300        OCT    20
      000463  000000000020    000   301        OCT    20
      000464  000000000020    000   302        OCT    20
      000465  000000000020    000   303        OCT    20
      000466  000000000020    000   304        OCT    20
      000467  000000000020    000   305        OCT    20
      000470  000000000020    000   306        OCT    20
      000471  000000000020    000   307        OCT    20
      000472  000000000020    000   308        OCT    20
      000473  000000000073    000   309        OCT    73
      000474  000000000020    000   310        OCT    20
      000475  000000000020    000   311        OCT    20
      000476  000000000020    000   312        OCT    20
```

Figure V.5. (Continued)

```
10018 01  02=11-71   19,109

      000477  000000000020    000   313        OCT    20
      000500  000000000020    000   314        OCT    20
      000501  000000000020    000   315        OCT    20
      000502  000000000020    000   316        OCT    20
      000503  000000000020    000   317        OCT    20
      000504  000000000020    000   318        OCT    20
      000505  000000000020    000   319        OCT    20
      000506  000000000020    000   320        OCT    20
      000507  000000000020    000   321        OCT    20
      000510  000000000020    000   322        OCT    20
      000511  000000000020    000   323        OCT    20
      000512  000000000020    000   324        OCT    20
      000513  000000000020    000   325        OCT    20
      000514  000000000020    000   326        OCT    20
      000515  000000000057    000   327        OCT    57
      000516  000000000075    000   328        OCT    75
      000517  000000000020    000   329        OCT    20
      000520  000000000020    000   330        OCT    20
      000521  000000000020    000   331        OCT    20
      000522  000000000020    000   332        OCT    20
      000523  000000000020    000   333        OCT    20
      000524  000000000020    000   334        OCT    20
      000525  000000000020    000   335        OCT    20
      000526  000000000020    000   336        OCT    20
      000527  000000000020    000   337        OCT    20
      000530  000000000020    000   338        OCT    20
      000531  000000000020    000   339        OCT    20
      000532  000000000020    000   340        OCT    20
      000533  000000000020    000   341        OCT    20
      000534  000000000020    000   342        OCT    20
      000535  000000000020    000   343        OCT    20
      000536  000000000020    000   344        OCT    20
      000537  000000000020    000   345        OCT    20
      000540  000000000020    000   346        OCT    20
      000541  000000000020    000   347        OCT    20
      000542  000000000020    000   348        OCT    20
      000543  000000000020    000   349        OCT    20
      000544  000000000020    000   350        OCT    20
      000545  000000000020    000   351        OCT    20
      000546  000000000020    000   352        OCT    20
      000547  000000000020    000   353        OCT    20
      000550  000000000020    000   354        OCT    20
      000551  000000000020    000   355        OCT    20
      000552  000000000020    000   356        OCT    20
      000553  000000000020    000   357        OCT    20
      000554  000000000020    000   358        OCT    20
      000555  000000000020    000   359        OCT    20
      000556  000000000020    000   360        OCT    20
      000557  000000000020    000   361        OCT    20
      000560  000000000020    000   362        OCT    20
      000561  000000000020    000   363        OCT    20
      000562  000000000020    000   364        OCT    20
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,109

        000563  000000000020    000  365        OCT   20
        000564  000000000020    000  366        OCT   20
        000565  000000000020    000  367        OCT   20
        000566  000000000020    000  368        OCT   20
        000567  000000000020    000  369        OCT   20
        000570  000000000020    000  370        OCT   20
        000571  000000000020    000  371        OCT   20
        000572  000000000020    000  372        OCT   20
        000573  000000000020    000  373        OCT   20
        000574  000000000020    000  374        OCT   20
        000575  000000000020    000  375        OCT   20
        000576  000000000020    000  376        OCT   20
        000577  000000000020    000  377        OCT   20
        000600  000000000020    000  378        OCT   20
        000601  000000000020    000  379        OCT   20
        000602  000000000020    000  380        OCT   20
        000603  000000000020    000  381        OCT   20
        000604  000000000020    000  382        OCT   20
        000605  000000000020    000  383        OCT   20
        000606  000000000020    000  384        OCT   20
        000607  000000000020    000  385        OCT   20
        000610  000000000020    000  386        OCT   20
        000611  000000000020    000  387        OCT   20
        000612  000000000020    000  388        OCT   20
        000613  000000000020    000  389        OCT   20
        000614  000000000020    000  390        OCT   20
        000615  000000000020    000  391        OCT   20
        000616  000000000020    000  392        OCT   20
        000617  000000000020    000  393        OCT   20
        000620  000000000020    000  394        OCT   20
        000621  000000000021    000  395        OCT   21
        000622  000000000022    000  396        OCT   22
        000623  000000000023    000  397        OCT   23
        000624  000000000024    000  398        OCT   24
        000625  000000000025    000  399        OCT   25
        000626  000000000026    000  400        OCT   26
        000627  000000000027    000  401        OCT   27
        000630  000000000030    000  402        OCT   30
        000631  000000000031    000  403        OCT   31
        000632  000000000020    000  404        OCT   20
        000633  000000000020    000  405        OCT   20
        000634  000000000020    000  406        OCT   20
        000635  000000000020    000  407        OCT   20
        000636  000000000020    000  408        OCT   20
        000637  000000000020    000  409        OCT   20
        000640  000000000020    000  410        OCT   20
        000641  000000000041    000  411        OCT   41
        000642  000000000042    000  412        OCT   42
        000643  000000000043    000  413        OCT   43
        000644  000000000044    000  414        OCT   44
        000645  000000000045    000  415        OCT   45
        000646  000000000046    000  416        OCT   46
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,109

    000647  000000000047    000   417          OCT     47
    000650  000000000050    000   418          OCT     50
    000651  000000000051    000   419          OCT     51
    000652  000000000020    000   420          OCT     20
    000653  000000000020    000   421          OCT     20
    000654  000000000020    000   422          OCT     20
    000655  000000000020    000   423          OCT     20
    000656  000000000020    000   424          OCT     20
    000657  000000000020    000   425          OCT     20
    000660  000000000020    000   426          OCT     20
    000661  000000000020    000   427          OCT     20
    000662  000000000062    000   428          OCT     62
    000663  000000000063    000   429          OCT     63
    000664  000000000064    000   430          OCT     64
    000665  000000000065    000   431          OCT     65
    000666  000000000066    000   432          OCT     66
    000667  000000000067    000   433          OCT     67
    000670  000000000070    000   434          OCT     70
    000671  000000000071    000   435          OCT     71
    000672  000000000020    000   436          OCT     20
    000673  000000000020    000   437          OCT     20
    000674  000000000020    000   438          OCT     20
    000675  000000000020    000   439          OCT     20
    000676  000000000020    000   440          OCT     20
    000677  000000000020    000   441          OCT     20
    000700  000000000000    000   442          OCT     0
    000701  000000000001    000   443          OCT     1
    000702  000000000002    000   444          OCT     2
    000703  000000000003    000   445          OCT     3
    000704  000000000004    000   446          OCT     4
    000705  000000000005    000   447          OCT     5
    000706  000000000006    000   448          OCT     6
    000707  000000000007    000   449          OCT     7
    000710  000000000010    000   450          OCT     10
    000711  000000000011    000   451          OCT     11
    000712  000000000020    000   452          OCT     20
    000713  000000000020    000   453          OCT     20
    000714  000000000020    000   454          OCT     20
    000715  000000000020    000   455          OCT     20
    000716  000000000020    000   456          OCT     20
    000717  000000000020    000   457          OCT     20
                       000720     458 IBM      BSS     8640
    021620  000001000477    000   459 CONTRL   OCT     000001000477
    021621  000476000301    000   460          OCT     000476000301
                       021622     461 GE       BSS     318
    022320  272520200600    000   462 HEADER   OCT     272520200600
    022321  002022634320    000   463          OCT     002022634320
    022322  272520473067    000   464          OCT     272520473067
    022323  200404050307    000   465          OCT     200404050307
    022324  200404050307    000   466          OCT     200404050307
    022325  202000000001    000   467          OCT     202000000001
    022326  200000000000    000   468          OCT     200000000000
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,109

      022327  202020000000    000   469         OCT     202020000000
      022330  263143252023    000   470         OCT     263143252023
      022331  462425200101    000   471         OCT     462425200101
      022332  000000000000    000   472         OCT     000000000000
      022333  000000000000    000   473         OCT     000000000000
      022334  000000000000    000   474         OCT     000000000000
      022335  000000000000    000   475         OCT     000000000000
      022336  202546262020    000   476 TRAIL   OCT     202546262020
      022337  000000000001    000   477         OCT     1
      022340  000000000003    000   478 FSKIP   OCT     3
      022341  000000000004    000   479 FREAD   OCT     4

     ERROR LINKAGE

      022342  000000000000    000
      022343  464725452020    000

        LITERALS

      022344  607777777777    000
      022345  000000003564    000
      022346  202546262020    000
      022347  000000000475    000
      022350  000475000000    000
                                    480         END
  22351 IS THE NEXT AVAILABLE LOCATION,   GMAP VERSION  JMPA/010271 JMPB/010271 JMPC/010271
  THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,111

OCTAL     SYMBOL    REFERENCES BY ALTER NO,

    71    ABORT3       59     48     59
    40    ABORT4       34     25     34
    15    ADVNCE       15     15     23     29
   157    AGAIN       113    108    113
   243    ALTER       165    162    165
    44    BEGIN        38     33     38     62    119
   320    CHARTB      202     97    202
   314    CHECK       198     82     86     87    198
    73    CONT         61     46     61
 21620    CONTRL      459    115    134    136    137    139    171    174    459
   247    DCW1        169     40    169
   255    DCW20       175      4    175
   251    DCW2        171    102    171
   252    DCW3        172     17    172
   253    DCW5        173    153    173
   254    DCW7        174    142    174
   171    EOT         123    110    123    131
   245    FA          167     17     40    167
   246    FB          168      4      9    102    125    142    148    153    158    168
   313    FILES       197     54     56    197
   312    FINI        196    128    130    161    196
 22341    FREAD       479     55    479
 22340    FSKIP       478     49     52     61    478
 21622    GE          461    133    187    461
    10    GEBORT             35     60    112    164    166
     1    GEINOS              2      7     15     38    100    123    140    146    151    156
     2    GEROAD              6     11     19     42    104    127    144    150    155    160
 22320    HEADER      462    175    462
   720    IBM         458     66    169    170    172    186    201    458
   202    IEOF        132     57    132
   316    INC         200    120    200
   305    INDEX       191     64    113    118    191
   310    LABELS      194     13     27     30     37    194
   311    LBLCNT      195     14     26     28    195
   304    LINT        190     63     71     72    117    190
   134    LOOP0        94     94    122
   136    LOOP1        96     96     99
   121    LOOP4        83     81     83     85     89
   303    NWDS        189     67     68     73     78     90    189
   307    OLDWDS      193     74     91    193
   315    ONE         199     84    114    199
     0    OPEN          2      1      2
   130    PASS         90     75     90
    14    RESET        14     14     53     58
    64    SKIPED       54     50     54
    42    START1       36     32     36
   256    STAT1       176     41     44     65    176
   274    STAT20      183      5    183
   260    STAT2       177    103    106    177
   262    STAT3       178     18     21    178
```

Figure V.5. (Continued)

```
10018 01  02-11-71   19,111

OCTAL     SYMBOL    REFERENCES BY ALTER NO,

   264    STAT4        179    10   149   159   179
   266    STAT5        180   126   180
   270    STAT7        181   143   181
   272    STAT8        182   154   182
   277    TALLY2       185    93    96   121   185
   300    TALLY3       186    92   186
   301    TALLY5       187    94   187
   302    TALLY6       188    95    98   188
   306    TIMES        192    77    88   192
 22336    TRAIL        476   116   145   173   476
   317    ZOUT1        201    80   201
**  17833 WORDS OF MEMORY WERE USED BY GMAP FOR THIS ASSEMBLY,
```

Figure V.5. (Concluded)

V.3 ABSTRACT-BIBLIOGRAPHY SEPARATION

The sub-section describes in detail the program GURGLE which separates the abstract information from the bibliographies. In the original concept of the concordance design GURGLE was to be used for each abstract before processing by the micro-concordance generator. However, it was found that computer utilization factors made it necessary to separate the processes.

V.3.1 Purpose

Routine GURGLE reads the raw transliterated data base text and updates the abstracts and bibliographic data pertaining to each abstract. Each is stored on separate tapes. By testing for the presence of certain delimiting fields and checking the relative order of their encounter, it is possible to separate accurately both fields.

V.3.2 Flowchart Comments for Program GURGLE (Keyed to Figure V.6)

1. Open all input and output files.
2. (It was found that each original APA tape occupied two reels upon transliteration of the text. GURGLE has been designed accordingly in order that all the documents for a particular year may be processed in one activity.)

   Set item SERIES to the status condition of V (FIRST) indicating that the tape designated as output file TAPER, file code 03, is to be processed first.
3. Present the NENT word of the tables ABSTORE and BIBLIO'STORE to 0 prior to isolation of the abstract and bibliographic data of a new document. Set status constant WHICH'TAPE to V(BIBLIO) prior to processing the next document. (The bibliographic data is encountered first.)
4. Set status item OPERAT to condition V(NORMAL) in the event that it had been previously set to condition V(DISASTER). This condition, V(DISASTER), is set if a delimiting field sequence error was encountered or isolated text buffer overflow occurs.

Initialize all variables used to isolate words, character, and delimiting field tests.

5. Read raw text from the proper input file into buffer ABSIN.

6. If an EOF is reached on either input file, transfer to close all input and output file and terminate job or reset to process second input tape.

   EOF TAPES? Yes => 88 No => 6

7. Isolate a word of text and store in item ONEWRD.

8. If the first word of the input buffer is 202546262020_8, item FINI, transfer to either set item SERIES to read second input tape or proceed to close all files and terminate the job.

   ONEWRD EQ 202546262020_8?

   No => 8 Yes => 88

9. Check for two consecutive words of all blanks.

   Condition met? Yes => 4 No => 9

10. Before normal character analysis is performed, test for status item OPERAT EQ V(DISASTER) indicating errors defined in FCC 3.

    OPERAT EQ V(DISASTER)?

    Yes => 76 No => 10

11. Shift the isolated word of the text and store bits 30-35 in item NXBYTE.

12. Construct the output file word by storing successive characters of ONEWRD in TPWRD. Increment item COUNT, the character counter.

13. If COUNT EQ 6, item TPWRD is completed. Proceed to store in either the abstract output buffer, ABSTORE, or the bibliographic data output buffer, BIBLIO'STORE.

    COUNT EQ 6?

    Yes => 13 No => 22

14. If status item WHICH'TAPE EQ V(ABSTRACT) transfer to code which generates the abstract output buffer. If not, proceed to code which generates the bibliographic data output buffer.

WHICH'TAPE EQ V(ABSTRACT)?

Yes => 14 No => 18

15. Enter TPWRD in ABSTORE and increment NENT (ABSTORE).

16. Test for buffer overflow (NENT(ABSTORE) 500).

    Buffer overflow? Yes => 16 No => 17

17. Set parameters for DISASTER mode in order to stream through the text until the delimiting field 01A is encountered, indicating a new document has been reached.

18. Zero-file TPWRD and reset COUNT to zero.

19. Enter TPWRD in BIBLIO'STORE and increment NENT (BIBLIO'STORE).

20. Test for buffer overflow (NENT(BIBLIO'STORE) GQ 500).

    Buffer overflow? Yes => 20 No => 21

21. Same as FCC 16.

22. Same as FCC 17.

23. A switch statement with 3 levels of transfer depending upon the value of item CHECK. This is a test for the field 02Z, indicating that the first half of the bibliographic data has been encountered.

24. NXBYTE EQ 0

    Yes => 24 No =>37

25. Increment item CHECK.

26. A switch statement with 9 levels of transfer depending upon which delimiting field is being tested:

    11Z: start of abstract

    12Z: end of abstract

    INC b̸: end of document

27. NXBYTE EQ 2?

    Yes => 28 No => 27

28. 0 => CHECK.

29. CHECK + 1 => CHECK.

30. Same as FCC 25.

31. NXBYTE EQ Z?

    Yes => 32 No => 31

32. 0 => CHECK.

33. 0 => CHECK.

34. Check for status item WHICH'TAPE EQ V(ABSTRACT) indicating a delimiting field sequence error has occurred and all variables must be reset.

    WHICH'TAPE EQ V(ABSTRACT)?

    Yes => 34 No => 35

35. V(BIBLIO) => WHICH'TAPE.

36. A two-level switch depending upon which portion of bibliographic data is being processed.

37. 1 => BIB'TEST.

38. Same as FCC 25.

39. NXBYTE EQ 1?

    Yes => 39 No => 85

40. FLAG+1 => FLAG.

41. NXBYTE EQ 1?

    Yes => 742 No => 41

42. 0 => FLAG 1.

43. FLAG1+1 6> FLAG1

44. NXBYTE EQ 2?

    Yes => 44 No => 46

45. Rotate TPWRD to left justify all characters.

46. Store TPWRD in BIBLIO'STORE and increment NENT (BIBLIO'STORE). Preset necessary variables and transfer to the limit of the loop to fetch the next character.

47. NXBYTE EQ 1?

    Yes => 85 No => 47

48. 0=7 FLAG 1.

49. NXBYTE EQ 1?
Yes => 49 No => 85

50. FLAG1+1 => FLAG1.

51. NXBYTE EQ 2?
Yes => 51 No => 52

52. FLAG+1 => FLAG.

53. 3 => FLAG1.

54. NXBYTE EQ Z?
Yes => 54 No => 58

55. Store TPWRD in ABSTORE and enter blanks, 0 (20)'s in unfilled bytes. Increment NENT (ABSTORE).

56. Test for abstract buffer overflow.

Buffer overflow?
Yes => 56 No => 57

57. Set OPERAT to V(DISASTER) and CHECK to 0. Transfer to stream through text in order to find next document.

58. Enter (000000000077$_8$) as the last word of the abstract. Reset a variable used to isolate text. Transfer to fetch the next character. Increment NENT (ABSTORE) and reset variables used to isolate characters. Set buffer indicator, WHICH'TAPE to condition V(BIBLIO).

59. 3 => FLAG 1.

60. NXBYTE EQ I?
Yes => 60 No => 85

61. FLAG1+1 => FLAG 1.

62. NXBYTE EQ N?
Yes => 63 No => 62

63. 6 => FLAG 1.

64. FLAG1+1 => FLAG1.

65. NXBYTE EQ C?

    Yes => 66 No => 65

66. 6 => FLAG 1.

67. FLAG1+1 => FLAG 1.

68. NXBYTE EQ 20_8? (blank)

    Yes => 69 No => 68

69. 6 => FLAG 1.

70. Shift the characters of the last word of the bibliographic to the left justified position and store in the bibliography buffer BIBLIO'STORE. Increment NENT (BIBLIO'STORE).

71. Test for bibliography buffer overflow.

    Buffer overflow? Yes => 71 No => 72

72. Set operation mode, item OPERAT, to V (DISASTER) and set parameters to stream through the text in order to search for the next document.

73. Enter 77_8 as the last word of the bibliographic data. Increment NENT (BIBLIO'STORE). Zero-fill the isolated word buffer, TPWRD, and reset the character count, item COUNT, to zero.

74. Write the isolated abstract and bibliography on output files ABSTRACT and BIBLIO, respectively.

75. Increment item DOC'TOTAL, the document counter.

76. Reset all parameters for the processing of the next document.

77. A three-level switch used to search for the character sequence 01A, indicating that a new document has been encountered.

78. NXBYTE EQ 0?

    Yes => 78 No => 85

79. CHECK+1 => CHECK.

80. NXBYTE EQ 1?

    Yes => 81 No => 82

81. 0 => CHECK.

82. CHECK+1 ⇒ CHECK?

83. NXBYTE EQ A?

Yes => 84 No => 83

84. 0 => CHECK.

85. Reset all parameters for the processing of the next document.

86. End of the loop which contains all code used to process each character of the isolated input buffer word, ONEWRD.

Loop completed?

Yes => 78 No => 9

87. End of loop which isolates each consecutive word from the input buffer and stores it in item ONEWRD.

Loop completed?

Yes => 4 No => 6

88. Check which tape was being read when the EOF was encountered. If it was the first tape, set item SERIES to V(SECOND) in order to process the second tape. If it was the second tape, transfer to close all files and terminate the job.

SERIES EQ V(FIRST)?

Yes => 89 No => 90

89. Set item SERIES to V(SECOND) and transfer to process second input tape.

90. Close all files and terminate job.

V.3.3 Summary of GURGLE Production

The time required to process 36254 abstracts was 1.4596 CPU hours.

Abstracts by Project Accession Number	RADC Tape # Abs./Biblio	Years
1-13825	45143/45142	67
13826-22493	45210/45214	68
22494-36254	45213/45212	69

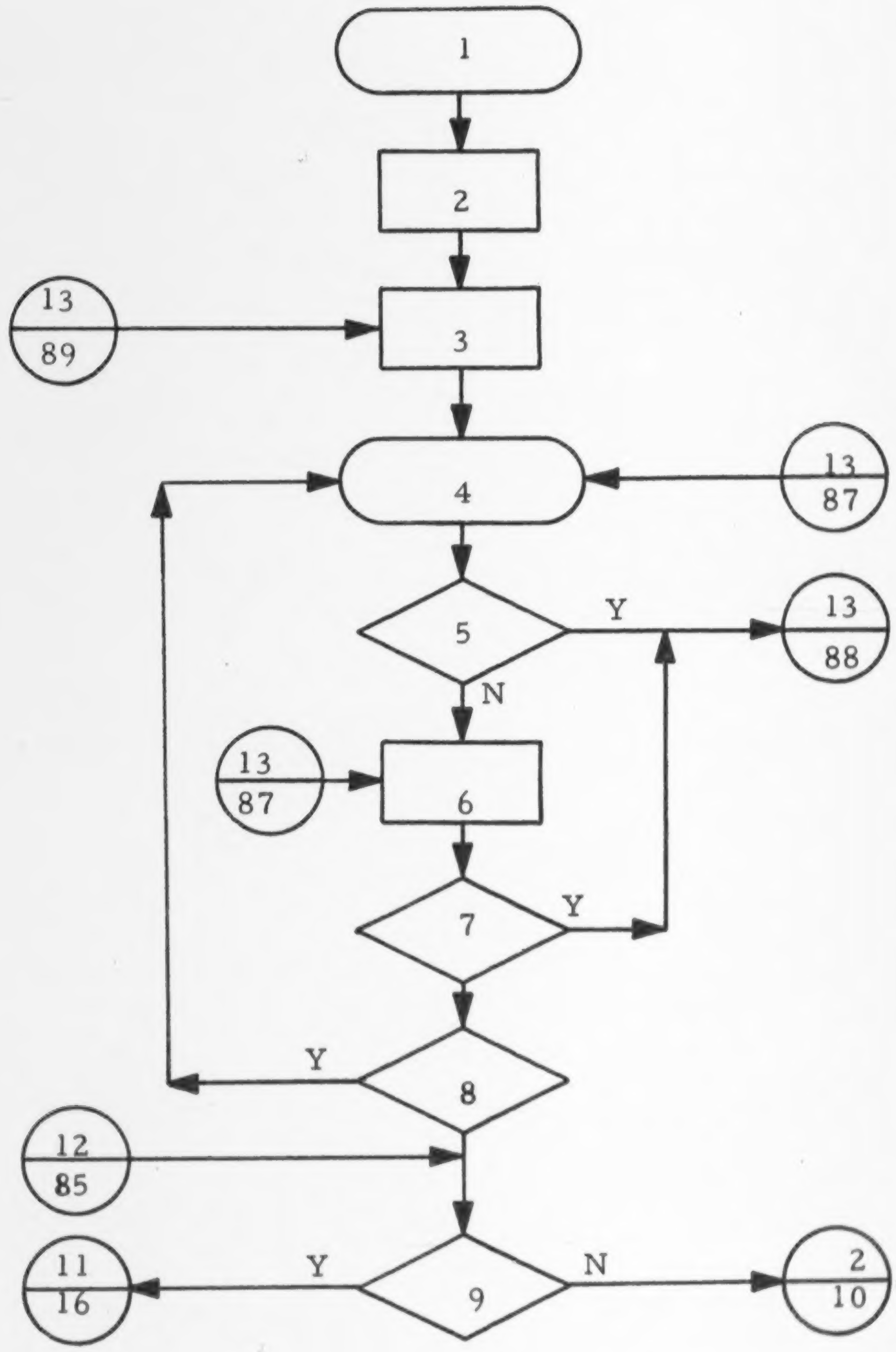

1
2
13
89
3
4
13
87
5
Y
13
88
N
13
87
6
7
Y
Y
8
12
85
Y
9
N
11
16
2
10

Figure V.6. GURGLE Flowchart

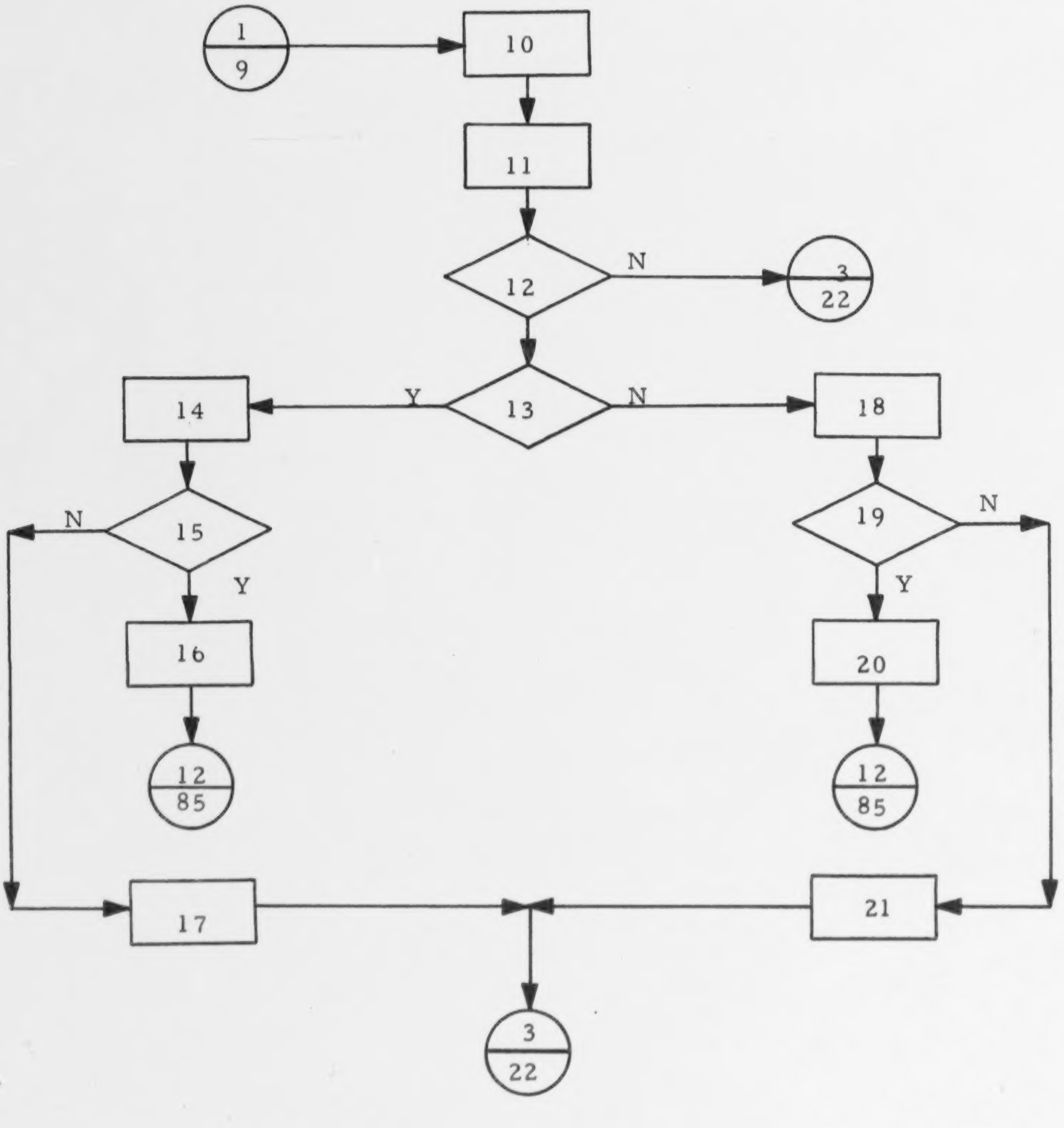

1
9
10
11
12
N
3
22
Y
13
N
14
18
N
15
Y
19
N
Y
16
20
12
85
12
85
17
21
3
22

Figure V.6. (Continued)

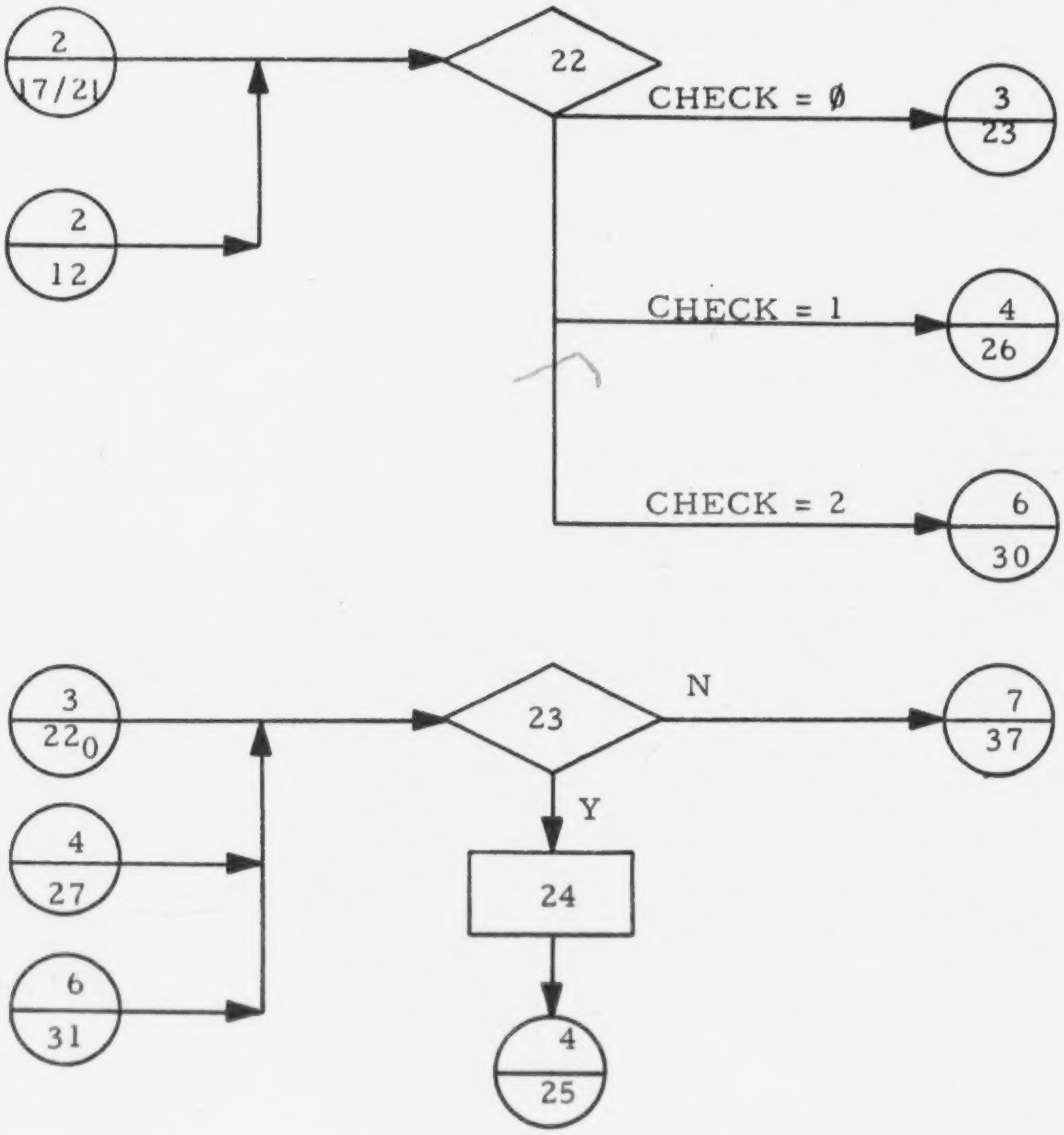

2
17/21
2
12
22
CHECK = 0
3
23
CHECK = 1
4
26
CHECK = 2
6
30
3
22$_0$
4
27
6
31
23
N
7
37
Y
24
4
25

Figure V. 6. (Continued)

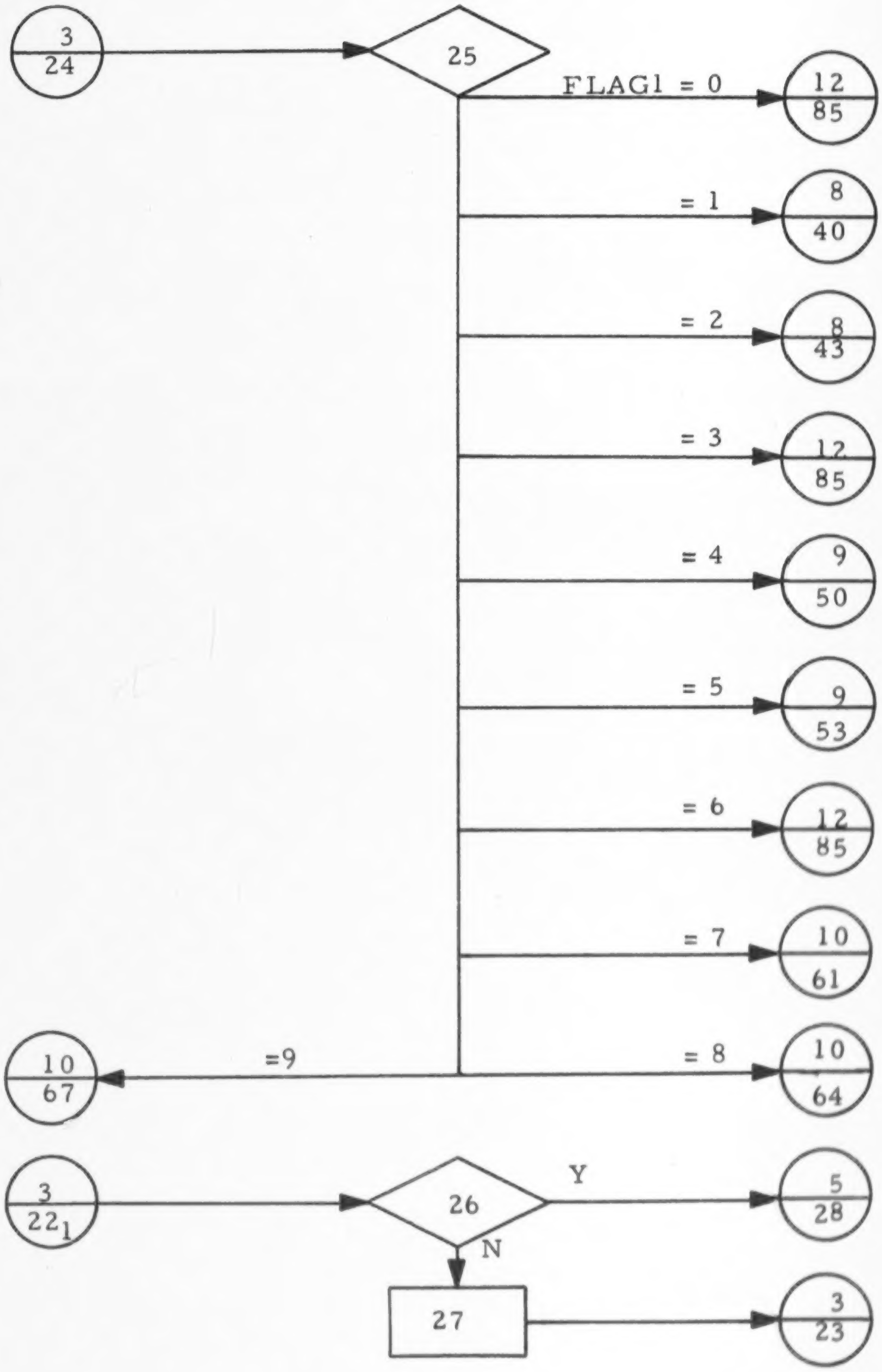

3
24
25
FLAG1 = 0
12
85
= 1
8
40
= 2
8
43
= 3
12
85
= 4
9
50
= 5
9
53
= 6
12
85
= 7
10
61
=9
10
67
= 8
10
64
3
22_1
26
Y
5
28
N
27
3
23

Figure V.6. (Continued)

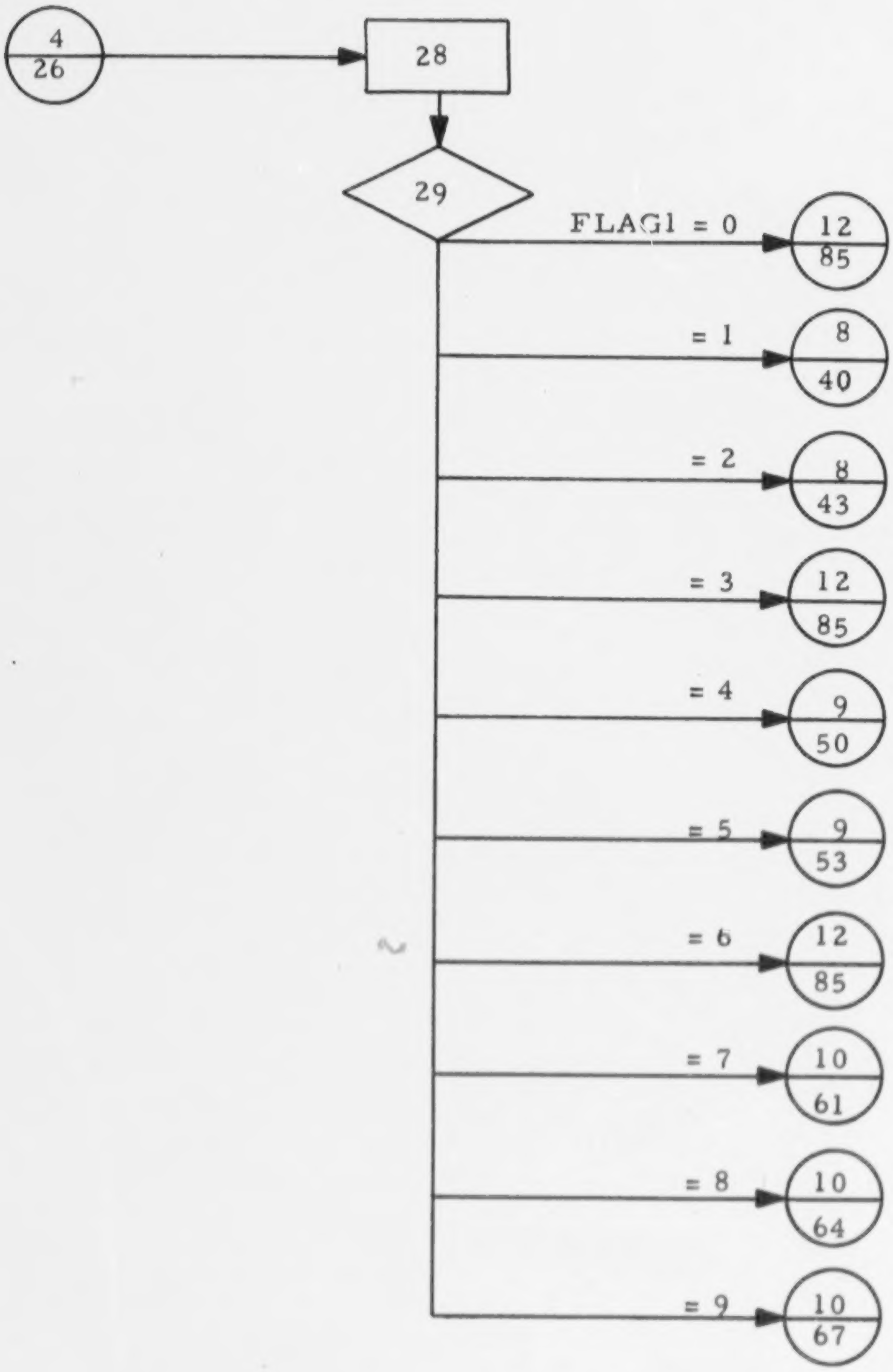

4
26
28
29
FLAG1 = 0
12
85
= 1
8
40
= 2
8
43
= 3
12
85
= 4
9
50
= 5
9
53
= 6
12
85
= 7
10
61
= 8
10
64
= 9
10
67

Figure V.6. (Continued)

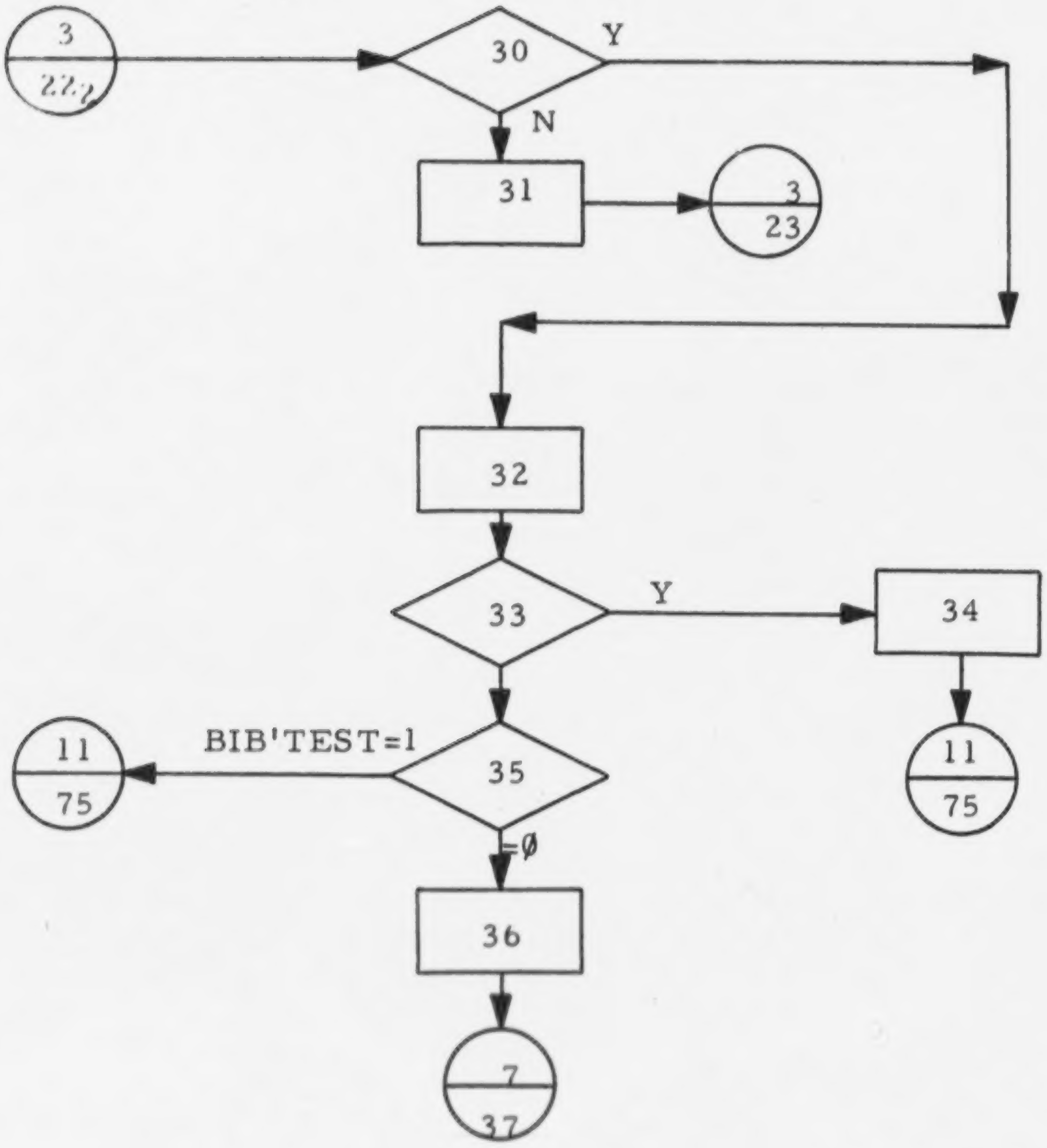

3
22
30
Y
N
31
3
23
32
33
Y
34
11
75
BIB'TEST=1
35
11
75
=0
36
7
37

Figure V.6. (Continued)

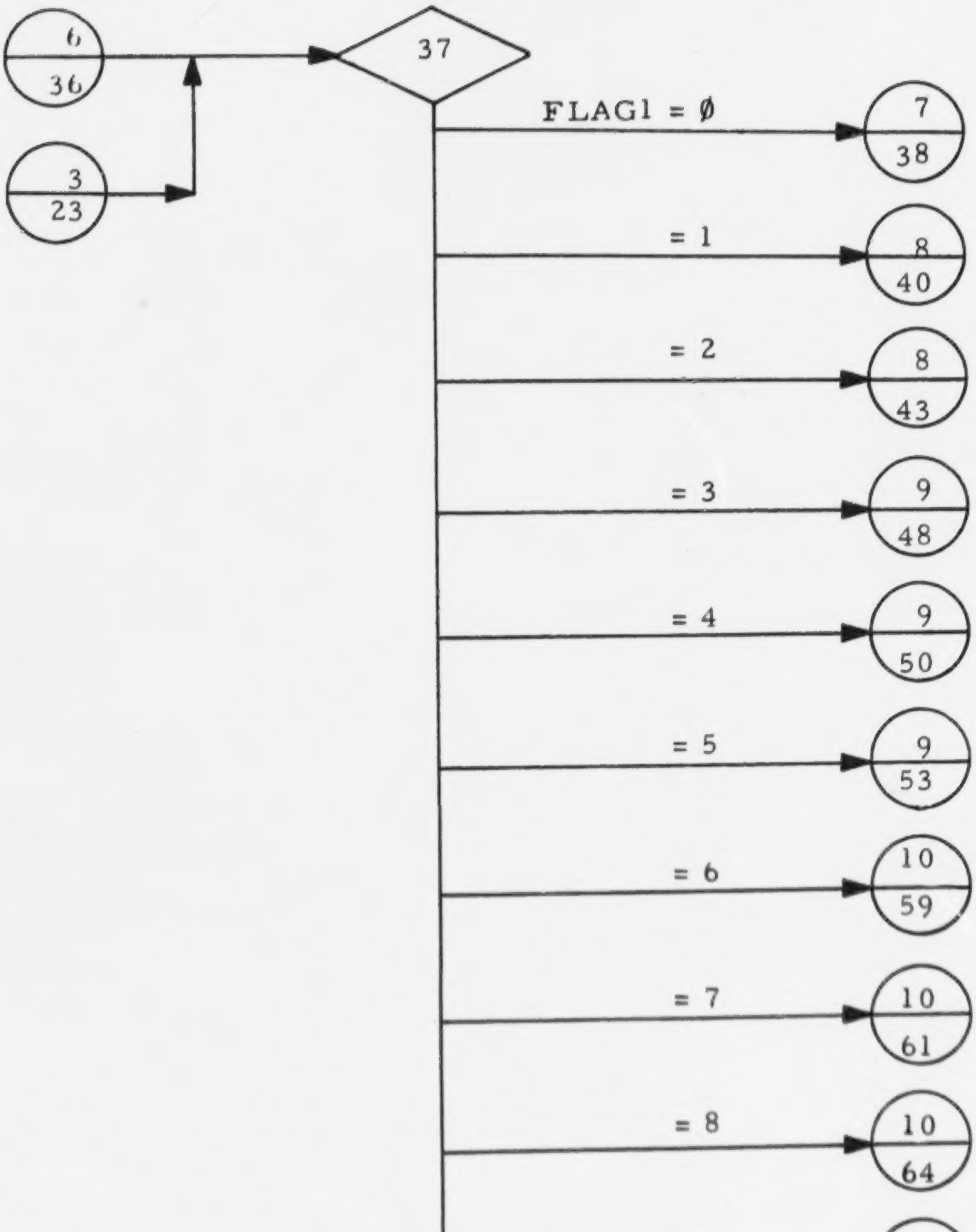

6
36
3
23
37
FLAG1 = Ø
7
38
= 1
8
40
= 2
8
43
= 3
9
48
= 4
9
50
= 5
9
53
= 6
10
59
= 7
10
61
= 8
10
64
= 9
10
67
7
37_0
38
N
Y
39
12
85

Figure V.6. (Continued)

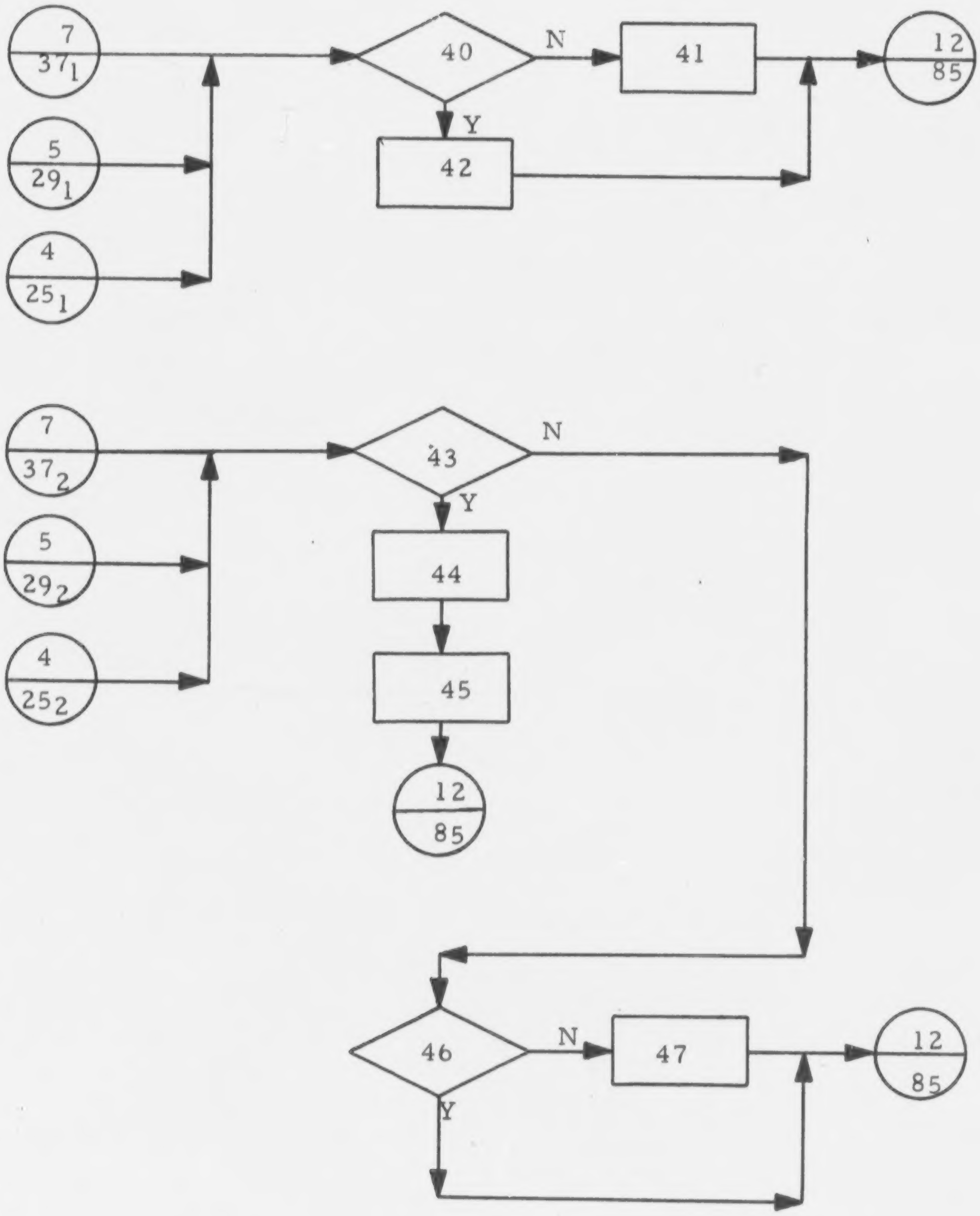

7
37_1
5
29_1
4
25_1
40
N
41
Y
42
12
85
7
37_2
5
29_2
4
25_2
43
N
Y
44
45
12
85
46
N
47
Y
12
85

Figure V.6. (Continued)

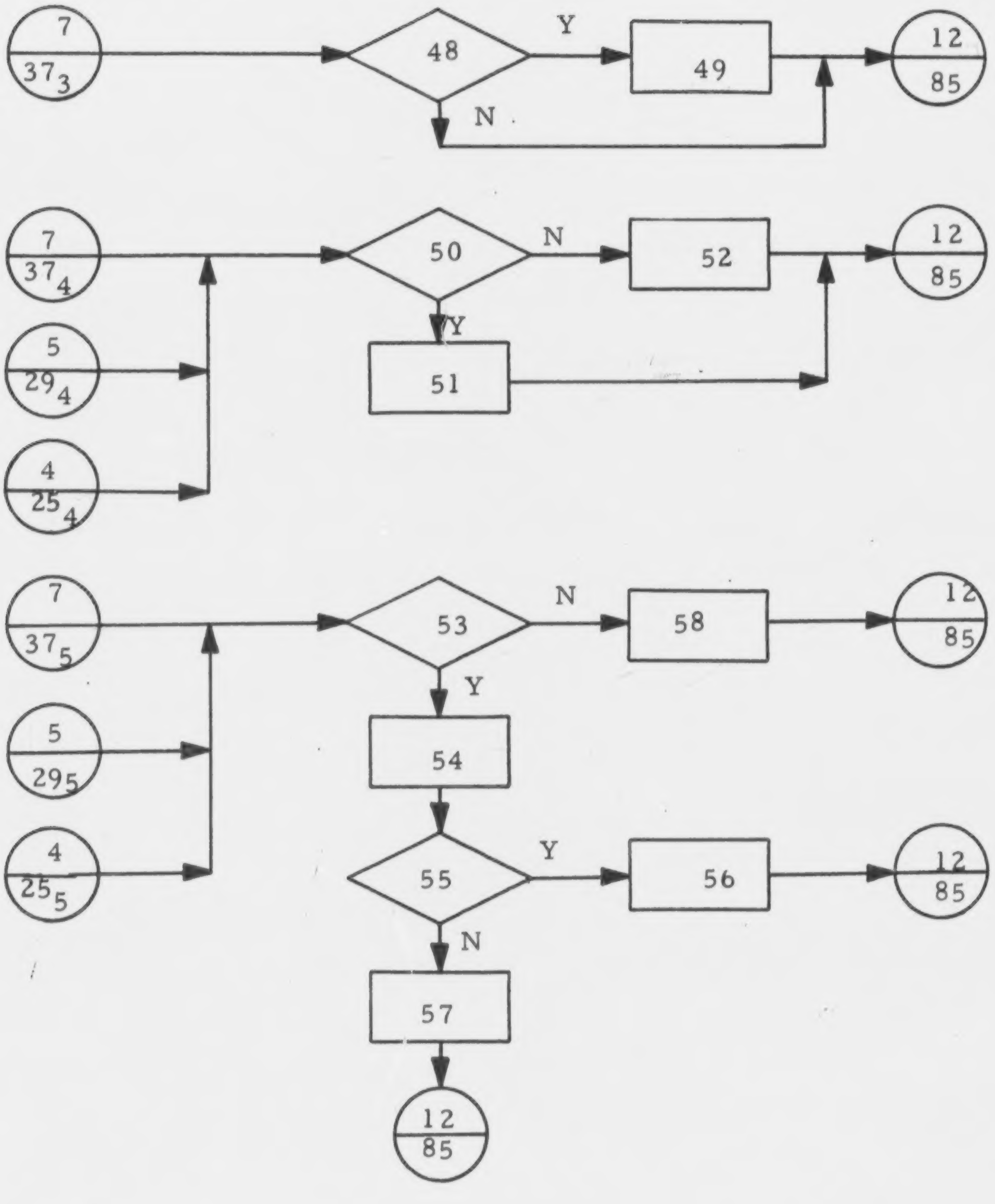

7
37_3
48
Y
49
N
12
85
7
37_4
5
29_4
4
25_4
50
N
52
Y
51
12
85
7
37_5
5
29_5
4
25_5
53
N
58
Y
54
55
Y
56
N
57
12
85
12
85
12
85

Figure V. 6. (Continued)

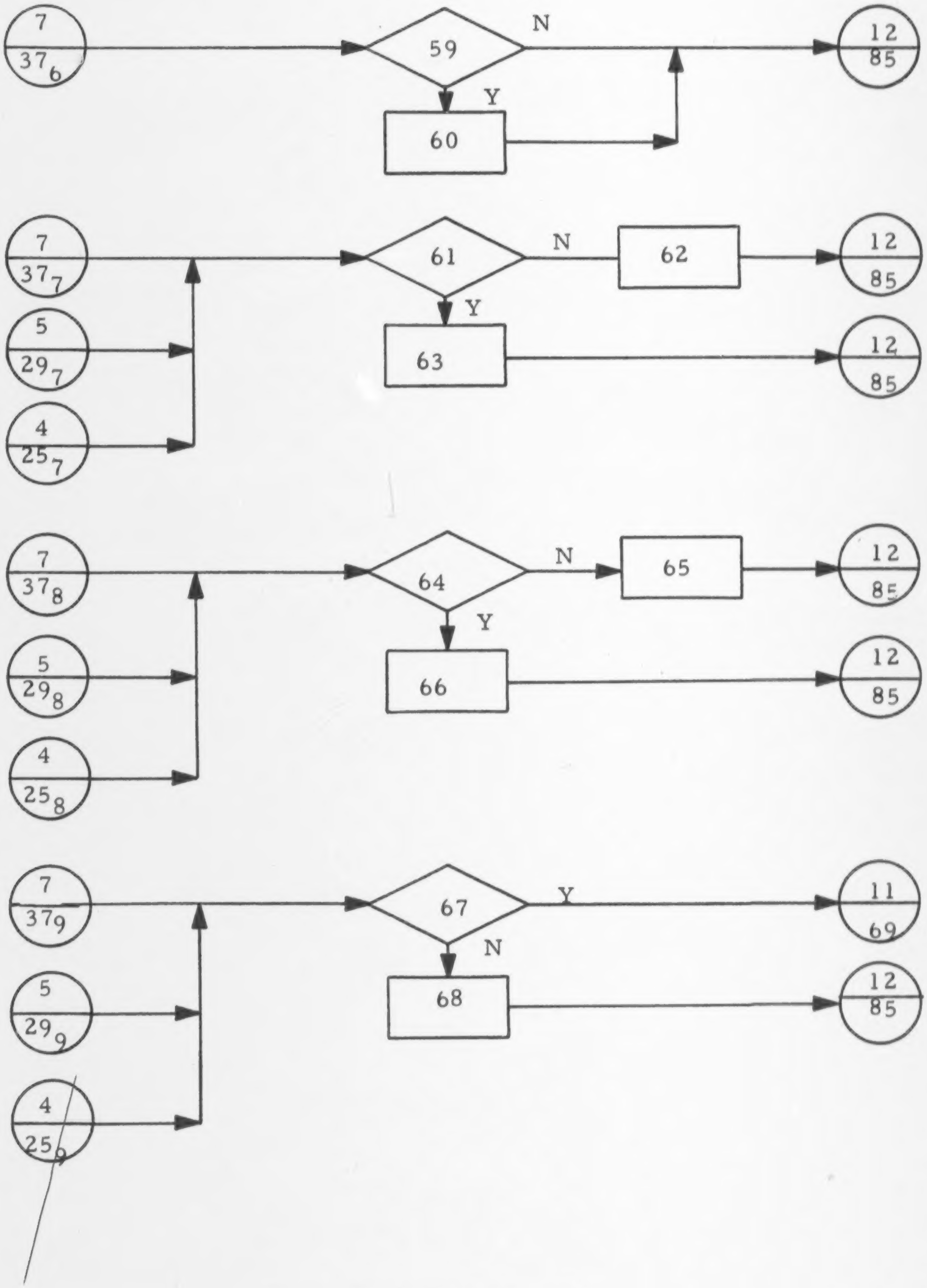

7
37_6
59
N
Y
60
12
85
7
37_7
5
29_7
4
25_7
61
N
62
Y
63
12
85
12
85
7
37_8
5
29_8
4
25_8
64
N
65
Y
66
12
85
12
85
7
37_9
5
29_9
4
25_9
67
Y
N
68
11
69
12
85

Figure V.6. (Continued)

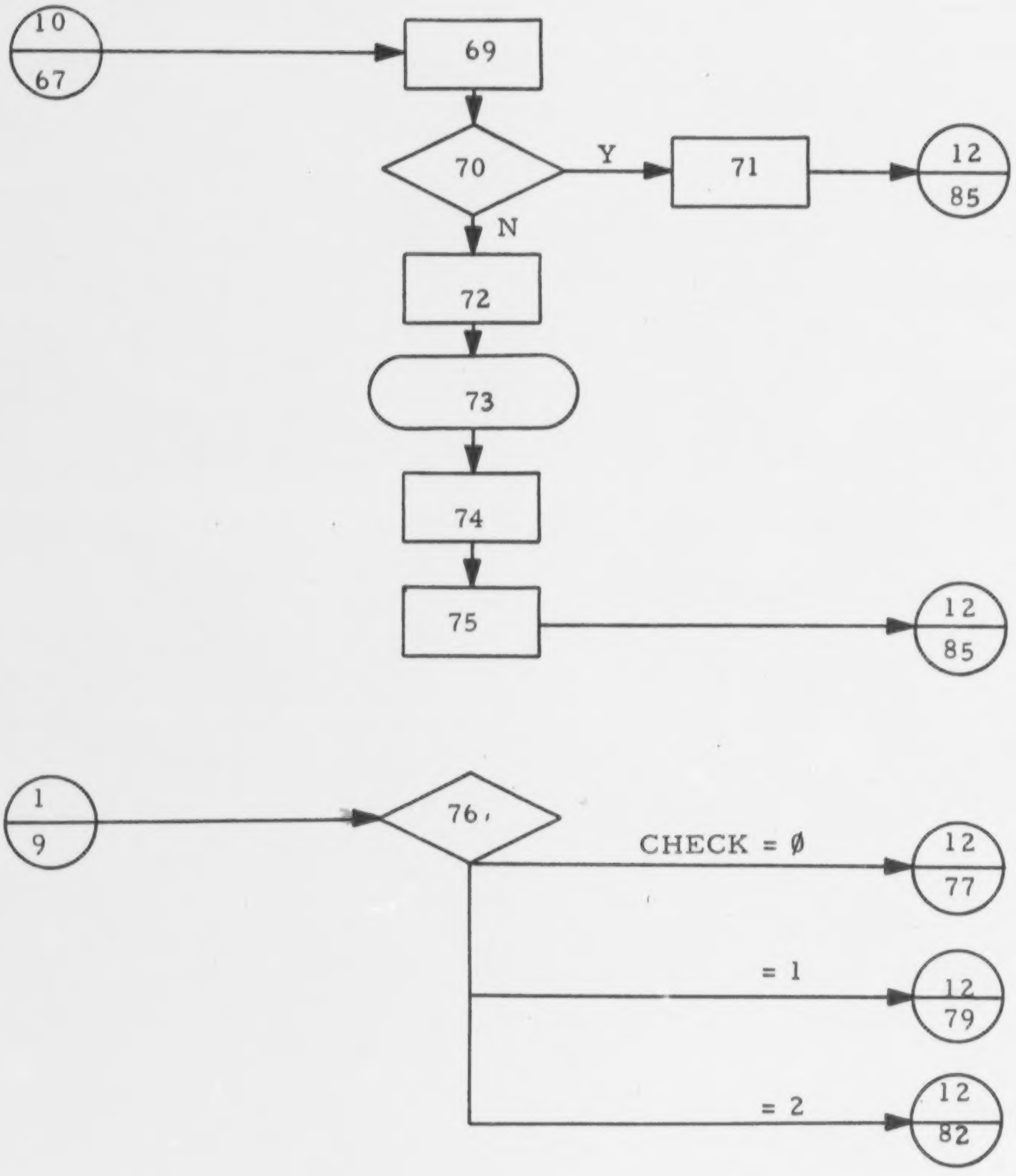

10
67
69
70
Y
71
12
85
N
72
73
74
75
12
85
1
9
76
CHECK = 0
12
77
= 1
12
79
= 2
12
82

Figure V.6. (Continued)

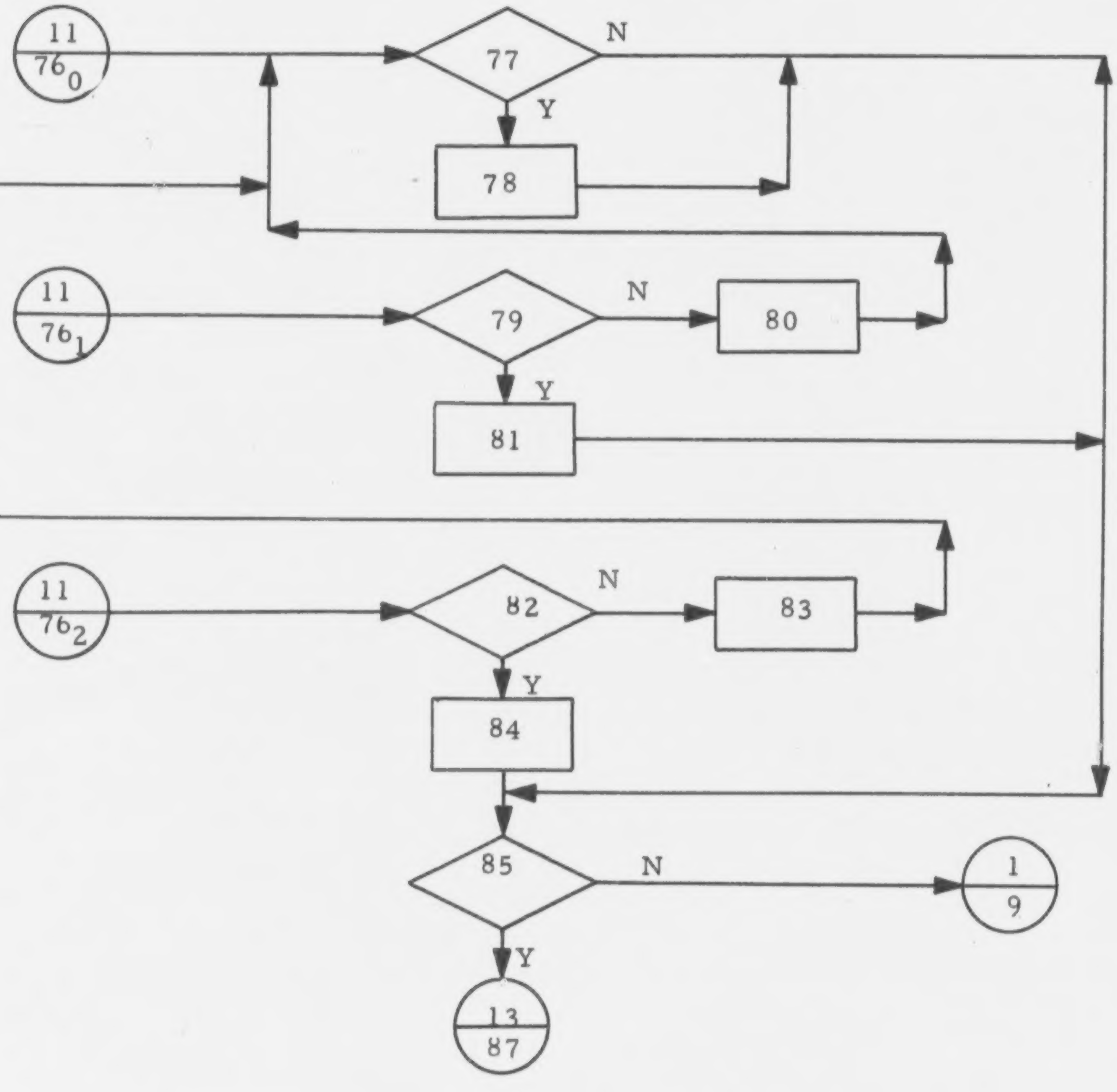

11
76_0
77
N
Y
78
11
76_1
79
N
80
Y
81
11
76_2
82
N
83
Y
84
85
N
1
9
Y
13
87

Figure V.6. (Continued)

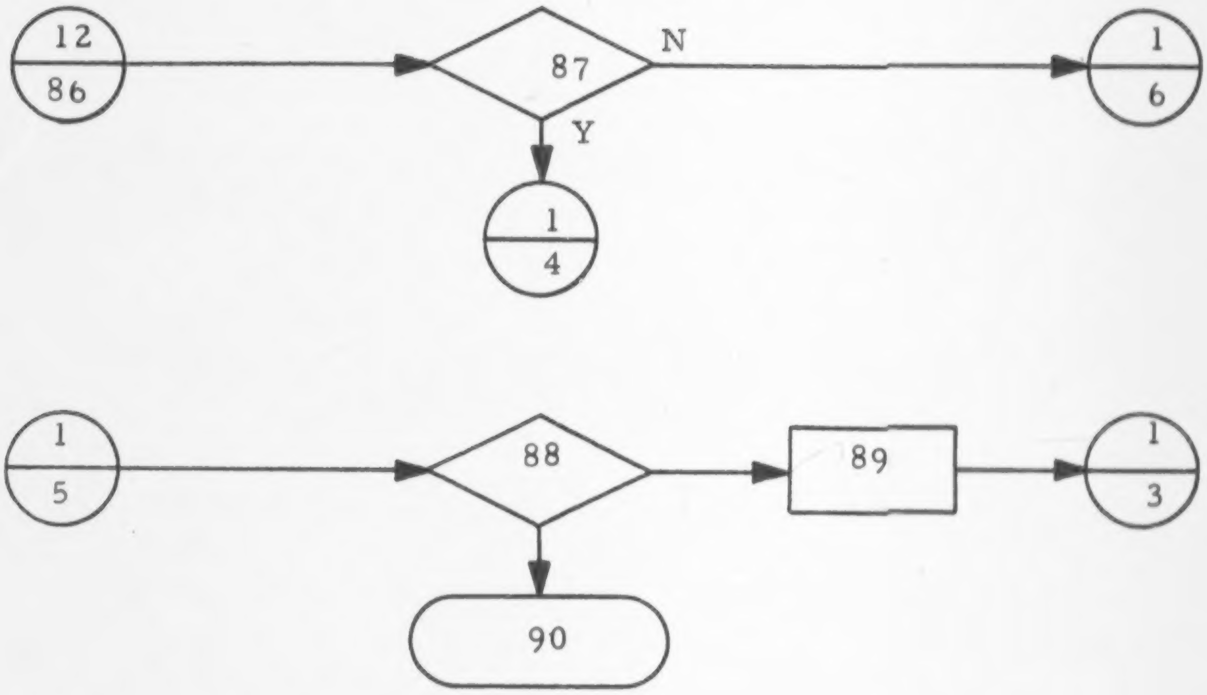

12
86
87
N
Y
1
6
1
4
1
5
88
89
1
3
90

Figure V.6. (Concluded)

```
21397 01  04-22-71   18,141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM GURGLE
  0000      START
  0001           TABLE ABSIN R 318 S 1 $
  0001            BEGIN ''ABSIN''
  0002             ITEM ABSIN'WRD H 6 0 0 $
  0003            END ''ABSIN''
  0003           TABLE ABSTORE V 500 S 1 $
  0003            BEGIN ''ABSTORE''
  0004             ITEM ABSTOR'WRD H 6 0 0 $
  0005            END ''ABSTORE''
  0005           TABLE BIBLIO'STORE V 500 S 1 $
  0005            BEGIN ''BIBLIO'STORE''
  0006             ITEM BIBLIO'WRD H 6 0 0 $
  0007            END ''BIBLIO'STORE''
  0007           ITEM OPERAT S V(DISASTER) V(NORMAL) $
  0008           ITEM WHICH'TAPE S V(BIBLIO) V(ABSTRACT) $
  0009           ITEM TALLY I 35 U P 0 $
  0010           ITEM FLAG1 I 35 U P 0 $
  0011           ITEM BIB'TEST I 35 U P 0 $
  0012           ITEM CHECK I 35 U P 0 $
  0013           ITEM NXBYTE H 1 $
  0014           ITEM TPWRD H 6 $
  0015           ITEM DOC'TOTAL I 35 U P 0 $
  0016           ITEM ONEWRD H 6 $
  0017           ITEM COUNT I 36 S P 0 $
  0018           ITEM MASK H 1 P O(77) $
  0019           ITEM BLANKS H 6 P O(202020202020) $
  0020           ITEM FINI H 6 P O(202546262020) $
  0021           ITEM DUMMY H 6 P 6H(000000) $
  0022           ITEM SERIES S V(FIRST) V(SECOND) $
  0023           SWITCH MODE1=(CHECK1,CHECK2,CHECK3) $
  0024           SWITCH MODE2=(CHECK4,GURG17) $
  0025           SWITCH MODE3=(GETOUT,GURG4,GURG5,GETOUT.GURG8,GURG9,GETOUT,
  0025                        GURG13,GURG14,GURG15) $
  0026           SWITCH MODE4=(FORM1,FORM2,FORM4) $
  0027           SWITCH MODE6=(GURG3,GURG4,GURG5,GURG7,GURG8,GURG9,GURG12,GURG13,
  0027                        GURG14,GURG15) $
  0028           FILE ABSTRACT V(NORM) V(NULL) V(SEQ) V(EOF) V(BUF) V(PAR) R01 $
  0029           FILE BIBLIO   V(NORM) V(NULL) V(SFG) V(EOF) V(BUF) V(PAR) R02 $
  0030           FILE TAPER V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR)  R03 $
  0031           FILE TAPES V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R04 $
  0032           FILE PRNT V(NORM) R06 $
  0033      GO.  GOTO GURGLE $
  0033      ''*************************************************************''
  0033      ''                         GURGLE                              ''
  0033      ''*************************************************************''
  0034           CLOSE GURGLE $
  0035            BEGIN ''GURGLE''
  0036             OUT(1,ABSTRACT) $
  0037             OUT(1,BIBLIO) $
```

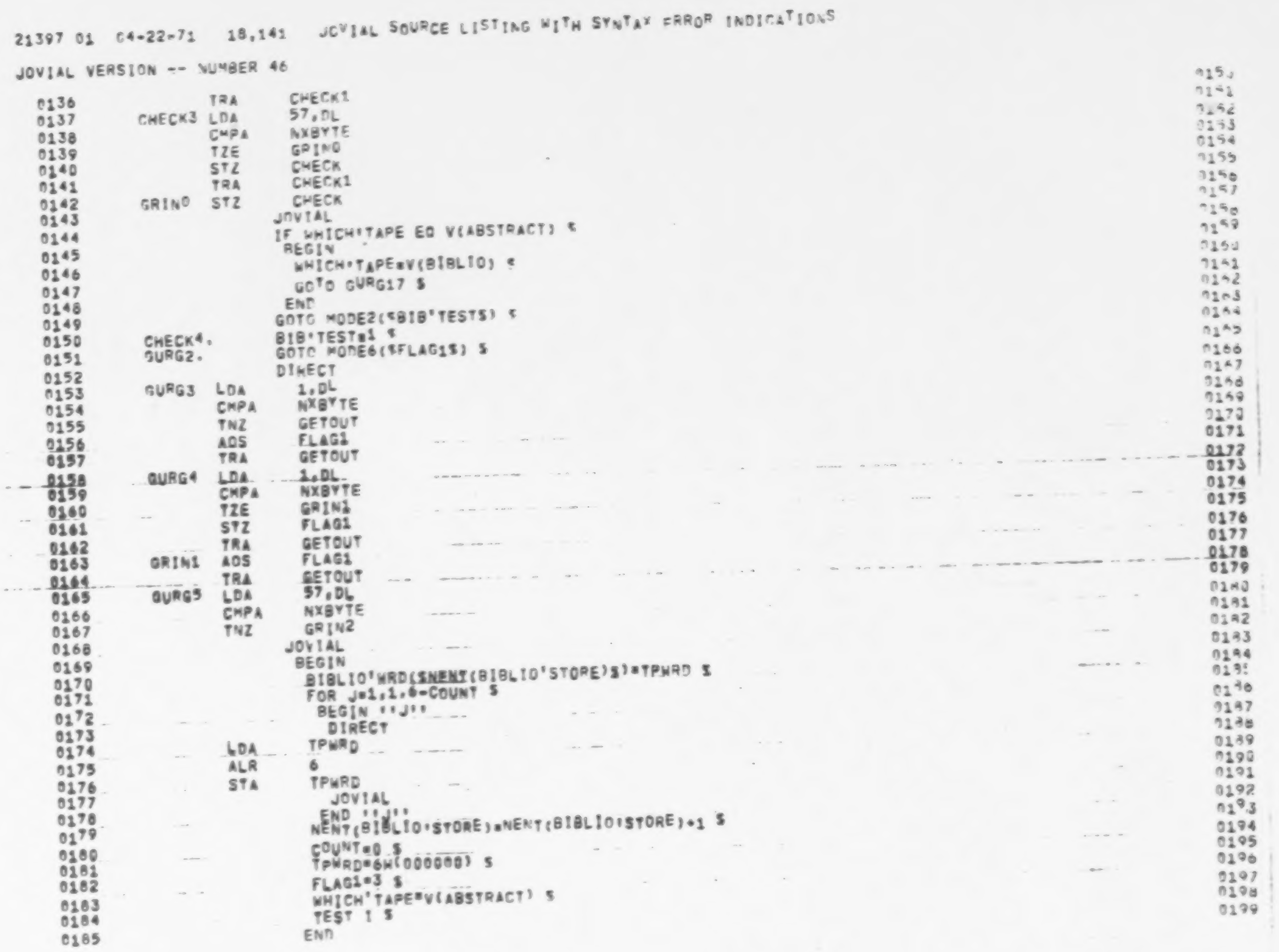
Figure V.7. GURGLE Listing

```
21397 01  04-22-71   18.141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0038                     IN(1,TAPER) $
 0039                     IN(1,TAPES) $
 0040                     OUT(1,PRNT) $
 0041                     SERIES=V(FIRST) $
 0042          GURGO.
 0042                     NENT(ABSTORE)=0 $
 0043                     NENT(BIBLIO'STORE)=0 $
 0044                     WHICH'TAPE=V(BIBLIO) $
 0045                     OPERAT=V(NORMAL) $
 0046                     COUNT=0 $
 0047                     TPWRD=6H(000000) $
 0048                     TALLY=0 $
 0049                     FLAG1=0 $
 0050                     BIB'TEST=0 $
 0051                     CHECK=0 $
 0052          GURG1.
 0052                     IF SERIES EQ V(SECOND) $
 0053                     GOTO SECOND $
 0054                     IO001(0,TAPER,0,ABSIN,318) $
 0055                     IF TAPER EQ V(EOF) $
 0056                     GOTO IEOF $
 0057                     GOTO GO1 $
 0058          SECOND.    IO009(0,TAPES,0,ABSIN,318) $
 0059                     IF TAPES EQ V(EOF) $
 0060                     GOTO IEOF $
 0061          GO1.       FOR N=ALL(ABSIN) $
 0062                      BEGIN ''N''
 0063                       ONEWRD=ABSIN'WRD($N$) $
 0064                       DIRECT
 0065                 SZN     N
 0066                 TNZ     *+4
 0067                 LDA     ONEWRD
 0068                 CMPA    FINI
 0069                 TZE     IEOF
 0070                 LDA     BLANKS
 0071                 CMPA    ONEWRD
 0072                 TNZ     GRIN10
 0073                 LDA     N
 0074                 LDQ     ABSIN+2,AL
 0075                 CMPQ    BLANKS
 0076                 TZE     GURG1
 0077                       JOVIAL
 0078          GRIN10.    FOR I=0,1,5 $
 0079                        BEGIN ''I''
 0080                         DIRECT
 0081                 SZN     OPERAT
 0082                 TZE     REFORM
 0083                 LDA     ONEWRD
 0084                 ALR     6
 0085                 STA     ONEWRD
```

Figure V.7. (Continued)

```
21397 01  04-22-71   18,141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

                ANA     MASK
                STA     NXBYTE
                LDQ     TPWRD
                QLR     6
                STQ     TPWRD
                STCA    TPWRD,01
                ADS     COUNT
                      JOVIAL
                      IF COUNT LS 6 $
                      GOTO CHECK0 $
                      IF WHICH'TAPE EQ V(ABSTRACT) $
                      GOTO GURG6 $
                      BIBLIO'WRD($NENT(BIBLIO'STORE)$)=TPWRD $
                      NENT(BIBLIO'STORE)=NENT(BIBLIO'STORE)+1 $
                      IF NENT(BIBLIO'STORE) GQ 500 $
                       BEGIN
                        CHECK=0 $
                        OPERAT=V(DISASTER) $
                        TEST I $
                       END
                      COUNT=0 $
                      TPWRD=6H(000000) $
                      GOTO CHECK0 $
        GURG6.        ABSTOR'WRD($NENT(ABSTORE)$)=TPWRD $
                      NENT(ABSTORE)=NENT(ABSTORE)+1 $
                      IF NENT(ABSTORE) GQ 500 $
                       BEGIN
                        CHECK=0 $
                        OPERAT=V(DISASTER) $
                        TEST I $
                       END
                      COUNT=0 $
                      TPWRD=6H(000000) $
        CHECK0.       GOTO MODE1($CHECK$) $
                      DIRECT
        CHECK1  LDA     0,DL
                CMPA    NXBYTE
                TNZ     GURG2
                AOS     CHECK
                      JOVIAL
                      GOTO MODE3($FLAG1$) $
                      DIRECT
        CHECK2  LDA     2,DL
                CMPA    NXBYTE
                TNZ     GRIN
                AOS     CHECK
                      JOVIAL
                      GOTO MODE3($FLAG1$) $
                      DIRECT
        GRIN    STZ     CHECK
```

Figure V.7. (Continued)



```
21397 01  04-22-71   18,141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0136              TRA     CHECK1                                                        0150
 0137      CHECK3  LDA     57,DL                                                         0151
 0138              CMPA    NXBYTE                                                        0152
 0139              TZE     GRIN0                                                         0153
 0140              STZ     CHECK                                                         0154
 0141              TRA     CHECK1                                                        0155
 0142      GRIN0   STZ     CHECK                                                         0156
 0143                     JOVIAL                                                         0157
 0144                     IF WHICH'TAPE EQ V(ABSTRACT) $                                 0158
 0145                      BEGIN                                                         0159
 0146                       WHICH'TAPE=V(BIBLIO) $                                       0160
 0147                       GOTO GURG17 $                                                0161
 0148                      END                                                           0162
 0149                     GOTO MODE2($BIB'TESTS) $                                       0163
 0150      CHECK4.        BIB'TEST=1 $                                                   0164
 0151      GURG2.         GOTO MODE6($FLAG1$) $                                          0165
 0152                     DIRECT                                                         0166
 0153      GURG3   LDA      1,DL                                                         0167
 0154              CMPA     NXBYTE                                                       0168
 0155              TNZ      GETOUT                                                       0169
 0156              AOS      FLAG1                                                        0170
 0157              TRA      GETOUT                                                       0171
 0158      GURG4   LDA      1,DL                                                         0172
 0159              CMPA     NXBYTE                                                       0173
 0160              TZE      GRIN1                                                        0174
 0161              STZ      FLAG1                                                        0175
 0162              TRA      GETOUT                                                       0176
 0163      GRIN1   AOS      FLAG1                                                        0177
 0164              TRA      GETOUT                                                       0178
 0165      GURG5   LDA      57,DL                                                        0179
 0166              CMPA     NXBYTE                                                       0180
 0167              TNZ      GRIN2                                                        0181
 0168                     JOVIAL                                                         0182
 0169                      BEGIN                                                         0183
 0170                       BIBLIO'WRD($NENT(BIBLIO'STORE)$)=TPWRD $                     0184
 0171                       FOR J=1,1,6-COUNT $                                          0185
 0172                        BEGIN ''J''                                                 0186
 0173                         DIRECT                                                     0187
 0174              LDA      TPWRD                                                        0188
 0175              ALR      6                                                            0189
 0176              STA      TPWRD                                                        0190
 0177                          JOVIAL                                                    0191
 0178                        END ''J''                                                   0192
 0179                       NENT(BIBLIO'STORE)=NENT(BIBLIO'STORE)+1 $                    0193
 0180                       COUNT=0 $                                                    0194
 0181                       TPWRD=6H(000000) $                                           0195
 0182                       FLAG1=3 $                                                    0196
 0183                       WHICH'TAPE=V(ABSTRACT) $                                     0197
 0184                       TEST 1 $                                                     0198
 0185                      END                                                           0199
```

Figure V.7. (Continued)

```
21397 01  04-22-71   18.141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

                DIRECT
GRIN2   LDA       1,DL
        CMPA      NXBYTE
        TZE       GETOUT
        STZ       FLAG1
        TRA       GETOUT
GURG7   LDA       1,DL
        CMPA      NXBYTE
        TNZ       GETOUT
        AOS       FLAG1
        TRA       GETOUT
GURG8   LDA       2,DL
        CMPA      NXBYTE
        TNZ       GRIN3
        AOS       FLAG1
        TRA       GETOUT
GRIN3   LDA       3,DL
        STA       FLAG1
        TRA       GETOUT
GURG9   LDA       57,DL
        CMPA      NXBYTE
        TNZ       GRIN4
                JOVIAL
                 BEGIN
                  ABSTOR'WRD($NENT(ABSTORE)$)=TPWRD $
                  IF COUNT LS 4 $
                  GOTO TERM1 $
                  FOR J=1,1,6-COUNT $
                   BEGIN ''J''
                    DIRECT
        LDA       TPWRD
        ALR       6
        STA       TPWRD
                    JOVIAL
                   END ''J''
                  IFEITH COUNT EQ 5 $
                  BYTE($2,4$)(ABSTOR'WRD($NENT(ABSTORE)$))=0(20202020) $
                  ORIF COUNT EQ 4 $
                  BYTE($1,5$)(ABSTOR'WRD($NENT(ABSTORE)$))=0(2020202020) $
                  END
                  GOTO TERM2 $
TERM1.            NENT(ABSTORE)=NENT(ABSTORE)-1 $
                  IFEITH COUNT EQ 0 $
                  BYTE($3,3$)(ABSTOR'WRD($NENT(ABSTORE)$))=0(202020) $
                  ORIF COUNT EQ 1 $
                  BYTE($4,2$)(ABSTOR'WRD($NENT(ABSTORE)$))=0(2020) $
                  ORIF COUNT EQ 2 $
                  BYTE($5,1$)(ABSTOR'WRD($NENT(ABSTORE)$))=0(20) $
                  END
TERM2.            COUNT=0 $
```

Figure V.7. (Continued)

```
21397 01  04-22-71   18.141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

                        NENT(ABSTORE)=NENT(ABSTORE)+1 $
                        IF NENT(ABSTORE) GQ 500 $
                         BEGIN
                          CHECK=0 $
                          OPERAT=V(DISASTER) $
                          TEST I $
                         END
                      ABSTOR'WRD($NENT(ABSTORE)$)=O(000000000077) $
                      NENT(ABSTORE)=NENT(ABSTORE)+1 $
                        TPWRD=6H(000000) $
                        FLAG1=6 $
                        WHICH'TAPE=V(BIBLIO) $
                        TEST I $
                       END
                      DIRECT
       GRIN4  LDA       3,DL
              STA       FLAG1
              TRA       GETOUT
       GURG12 LDA       25,DL
              CMPA      NXBYTE
              TNZ       GETOUT
              AOS       FLAG1
              TRA       GETOUT
       GURG13 LDA       37,DL
              CMPA      NXBYTE
              TZE       GRIN5
              LDA       6,DL
              STA       FLAG1
              TRA       GETOUT
       GRIN5  AOS       FLAG1
              TRA       GETOUT
       GURG14 LDA       19,DL
              CMPA      NXBYTE
              TZE       GRIN6
              LDA       6,DL
              STA       FLAG1
              TRA       GETOUT
       GRIN6  AOS       FLAG1
              TRA       GETOUT
       GURG15 LDA       16,DL
              CMPA      NXBYTE
              TZE       GRIN7
              LDA       6,DL
              STA       FLAG1
              TRA       GETOUT
                      JOVIAL
       GRIN7.         IF COUNT EQ 0 $
                      GOTO GRIN11 $
                      BIBLIO'WRD($NENT(BIBLIO'STORE)$)=TPWRD $
                      FOR J=1,1,6-COUNT $
```

Figure V.7. (Continued)

```
21397 01  04-22-71   18,141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0286                      BEGIN ''J''                                                      0300
 0287                       DIRECT                                                          0301
 0288             LDA       TPWRD                                                           0302
 0289             ALR       6                                                               0303
 0290             STA       TPWRD                                                           0304
 0291                       JOVIAL                                                          0305
 0292                      END ''J''                                                        0306
 0293                     NENT(BIBLIO'STORE)=NENT(BIBLIO'STORE)+1 $                         0307
 0294                     IF NENT(BIBLIO'STORE) GO 500 $                                    0308
 0295                      BEGIN                                                            0309
 0296                       CHECK=0 $                                                       0310
 0297                       OPERAT=V(DISASTER) $                                            0311
 0298                       TEST I $                                                        0312
 0299                      END                                                              0313
 0300                     BIBLIO'WRD($NENT(BIBLIO'STORE)$)=O(000000000077) $                0314
 0301                     NENT(BIBLIO'STORE)=NENT(BIBLIO'STORE)+1 $                         0315
 0302                     COUNT=0 $                                                         0316
 0303                     TPWRD=6H(000000) $                                                0317
 0304       GRIN11.       BIB'TEST=0 $                                                      0318
 0305       GURG16.       W1=NENT(ABSTORE) $                                                0319
 0306                     W2=NENT(BIBLIO'STORE) $                                           0320
 0307                     IO006(4,ABSTRACT,0,ABSTORE,W1) $                                  0321
 0308                     IO007(4,BIBLIO,0,BIBLIO'STORE,W2) $                               0322
 0309                     DOC'TOTAL=DOC'TOTAL+1 $                                           0323
 0310       GURG17.       TALLY=0 $                                                         0324
 0311                     COUNT=0 $                                                         0325
 0312                     TPWRD=6H(000000) $                                                0326
 0313                     FLAG1=0 $                                                         0327
 0314                     NENT(ABSTORE)=0 $                                                 0328
 0315                     NENT(BIBLIO'STORE)=0 $                                            0329
 0316                     IF BIB'TEST EQ 0 $                                                0330
 0317                     TEST I $                                                          0331
 0318                     BIB'TEST=1 $                                                      0332
 0319                     TEST I $                                                          0333
 0320       REFORM.       GOTO MODE4($CHECK$) $                                             0334
 0321                      DIRECT                                                           0335
 0322       FORM1   LDA      0,DL                                                           0336
 0323               CMPA     NXBYTE                                                         0337
 0324               TNZ      GETOUT                                                         0338
 0325               AOS      CHECK                                                          0339
 0326               TRA      GETOUT                                                         0340
 0327       FORM2   LDA      1,DL                                                           0341
 0328               CMPA     NXBYTE                                                         0342
 0329               TZE      FORM3                                                          0343
 0330               STZ      CHECK                                                          0344
 0331               TRA      FORM1                                                          0345
 0332       FORM3   AOS      CHECK                                                          0346
 0333               TRA      GETOUT                                                         0347
 0334       FORM4   LDA      17,DL                                                          0348
 0335               CMPA     NXBYTE                                                         0349
```

Figure V.7. (Continued)

```
21397 01  04=22-71  =18,141   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0336             TZE     FORM5                                                        0350
  0337             STZ     CHECK                                                        0351
  0338             TRA     FORM1                                                        0352
  0339      FORM5  STZ     TALLY                                                        0353
  0340             STZ     COUNT                                                        0354
  0341             STZ     TPWRD                                                        0355
  0342             STZ     FLAG1                                                        0356
  0343             STZ     CHECK                                                        0357
  0344             LDA     1,DL                                                         0358
  0345             STA     OPERAT                                                       0359
  0346                   JOVIAL                                                         0360
  0347                   NENT(ABSTORE)=0 $                                              0361
  0348                   NENT(BIBLIO'STORE)=0 $                                         0362
  0349                   BIB'TEST=0 $                                                   0363
  0350                   WHICH'TAPE=V(BIBLIO) $                                         0364
  0351      GETOUT.     END ''I''                                                       0365
  0352                END ''N''                                                         0366
  0353               GOTO GURG1 $                                                       0367
  0354      IEOF.                                                                       0368
  0354               IF SERIES EQ V(FIRST) $                                            0369
  0355                BEGIN                                                             0370
  0356                 SERIES=V(SECOND) $                                               0371
  0357                 GOTO GURG0 $                                                     0372
  0358                END                                                               0373
  0359               OUT(2,ABSTRACT) $                                                  0374
  0360               OUT(2,BIBLIO) $                                                    0375
  0361               IN(2,TAPER) $                                                      0376
  0362               IN(2,TAPES) $                                                      0377
  0363               IO999(3,PRNT,0,DOC'TOTAL,1) $                                      0378
  0364               OUT(2,PRNT) $                                                      0379
  0365              END ''GURGLE''                                                      0380
  0366      STOP $                                                                      0381
  0367      TERM GO $                                                                   0382
```

Figure V.7. (Concluded)

V.4 MICRO-CONCORDANCE GENERATION

This sub-section contains the documentation of the micro-concordance generator, including flowcharts, listings, procedure descriptions, method of operation, formats of input-output tapes, and instructions for use. Because of the inefficiency of JOVIAL-generated code in certain areas (notably character-handling), the time-dependent portions of the program were written in-line in GMAP (assembly). These sections are documented in greater detail than the rest of the program.

V.4.1 Design Considerations for the Micro-Concordance Generator

The micro-concordance generator fulfills two functions in the batch system:

1) Prepares output for the statistical filter
2) Prepares output for the document concept vector generator

Since these two packages have completely compatible requirements, there was no difficulty in creating magnetic tape files usable by both without modification.

Although brief reference was made to a definition of micro-concordance in Section II. 1 of this report, it is prudent to go into greater detail at this point. A concordance of a document collection is a list of all tokens in the text with a count of frequency of occurrence and some indication of place of occurrence with the collection. Because of the nature of the statistical measures used in the separation of contents from non-content stems (c.f. Section V.5), two kinds of concordances are needed for this collection:

1) A concordance of each document (a micro-concordance)
2) The master concordance of the whole collection

In addition, the raw token counts for each document are needed by the "statistical filtration" routines for normalization of the frequency of occurrence of tokens. Termed the "document leaders", the raw token counts are separated from the other concordance data, as the document concept vector generator does not need this information.

The design of the micro-concordance generator had to take another factor into consideration besides generating the correct data for the following programs: speed. The entire data base had to be passed through the micro-concordance code so that considerable attention was paid to time factors, the result of which is a fair amount of coding in GMAP, especially the character-handling portions. Maximum use was made of the available GESORT package, in order to decrease running times and reduce the man-hour effort to accomplish the production.

V.4.2 Implementation Details

The micro-concordance generator consists of two major phases: set-up and run. During the set-up phase the initial operating conditions are established; storage is cleared, tables of common words, suffixes, and character status are built for use during the second phase. Program operation terminates abnormally if any fatal errors occur during phase one. Phase two performs the actual micro-concordance generation, using the environment established by phase one.

Phase One

Set-up phase consists of four JOVIAL "CLOSE" routines: PRERUN, SMEAR, PLETAB, and SUFTAB, each one associated with one set-up function.

1. PRERUN:

   Function: To create a 64 word table indicating the status of every character in the HISI character set, against which every character in the input data is checked.

Method: The table PEEKTB is indexed by the numeric value of each character. Characters fall into four classes:

a. "break": used to indicate end-of-word condition

b. "ignore": completely ignored during concordance generation

c. "legal": characters to be used during concordance

d. "end-of-document": used as the "escape character" to indicate the end of an input unit, an abstract in this case

Operation: PRERUN operates in five discrete sections:

1. PEEKTB is first initialized with a "TRA" instruction to the program point in WORD which processes legal characters. In this way the legal character list (by far the largest) does not need to be explicitly prepared.

2. The position in PEEKTB for blank is initialized to the TRA instruction to the location in WORD which processes an isolated piece of text (blank is the default break character).

3. A list of "ignore" characters, one to a column, is read in from the standard input file, and is processed into the PEEKTB. If no ignore characters are listed, use a blank card.

4. A list of "break" characters, one to a column, is read in from the standard input file, and is processed into the PEEKTB. If a blank is desired as the only break character, use a blank card.

5. The position corresponding to the ! character (octal 77) is set to a TRA to the program step in WORD which completes the processing for a completely concorded abstract.

Figure V. 8 illustrates the symbolic layout of PEEKTB after processing by PRERUN. Of course, during actual execution of the program, PEEKTB will contain the true binary form of the TRA instructions along with the correct absolute addresses for ENDWRD, WORD1, MOVEBY, and PROCES. Figure V. 9 is the detailed flowchart for PRERUN. If either list of characters (break or ignore) is missing, this is a fatal error, and the run stops after printing an appropriate message. The two lists are punched one character to a column with NO embedded blanks allowed.

0	TRA	ENDWRD	↑	TRA	MOVEBY
1	TRA	ENDWRD	J		
2	TRA	ENDWRD	K		
3	TRA	ENDWRD	L	etc.	
4	TRA	ENDWRD	M		
5	TRA	ENDWRD	N		
6	TRA	ENDWRD	O		
7	TRA	ENDWRD	P		
8	TRA	ENDWRD	Q		
9	TRA	ENDWRD	R		
[	TRA	MOVEBY	-	TRA	ENDWRD
#			$	TRA	ENDWRD
@	etc.		*	TRA	ENDWRD
:			)	TRA	ENDWRD
>			;		
?			'	TRA	ENDWRD
b̸	TRA	WORD1	+	TRA	ENDWRD
A	TRA	MOVEBY	/	TRA	ENDWRD
B			S	TRA	MOVEBY
C			T		
D			U	etc.	
E	etc.		V		
F			W		
G			X		
H			Y		
I			Z		
&			←		
.	TRA	ENDWRD	,	TRA	ENDWRD
]			%		
(	TRA	ENDWRD	=	TRA	ENDWRD
<			"		
\			!	TRA	PROCES

Figure V.8. PEEKTB Symbolic Layout

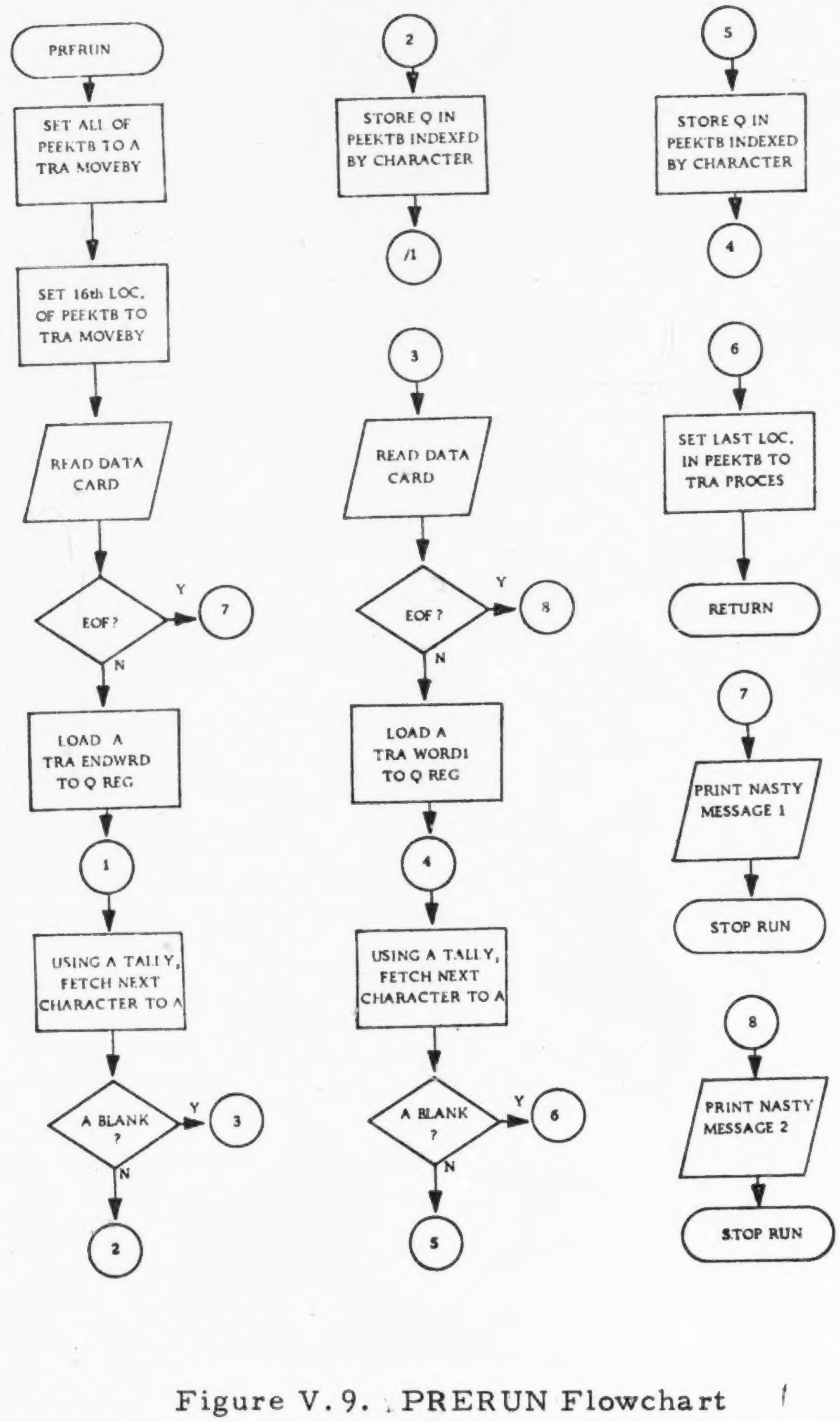

PRERUN
SET ALL OF PEEKTB TO A TRA MOVEBY
SET 16th LOC. OF PEEKTB TO TRA MOVEBY
READ DATA CARD
EOF?
Y
7
N
LOAD A TRA ENDWRD TO Q REG
1
USING A TALLY, FETCH NEXT CHARACTER TO A
A BLANK ?
Y
3
N
2
2
STORE Q IN PEEKTB INDEXED BY CHARACTER
1
3
READ DATA CARD
EOF ?
Y
8
N
LOAD A TRA WORD1 TO Q REG
4
USING A TALLY, FETCH NEXT CHARACTER TO A
A BLANK ?
Y
6
N
5
5
STORE Q IN PEEKTB INDEXED BY CHARACTER
4
6
SET LAST LOC. IN PEEKTB TO TRA PROCES
RETURN
7
PRINT NASTY MESSAGE 1
STOP RUN
8
PRINT NASTY MESSAGE 2
STOP RUN

Figure V.9. PRERUN Flowchart

2. SMEAR:

Function:

a. To zero out the following tables:
   1) COMWD1 (common words of one machine word (6 characters or less) in length
   2) COMWD2 (common words of two machine words (more than 6 characters, less than or equal to 12 characters) in length
   3) SUF1TB suffixes of one character
   4) SUF2TB suffixes of two characters
   5) SUF3TB suffixes of three characters
   6) SUF4TB suffixes of four characters
   7) ROOTS concorded roots (stems)
   8) WGHT weights of concorded roots

b. To set to "+1" vowel positions by actual value of the vowel letters A, E, I, O, U in the table ELIM, for use in stem analysis

Method: "FOR" loops referencing the table locations; code is self-explanatory.

Figure V.10 is the flowchart for SMEAR.

3. PLETAB:

Function: To create two tables of hash-addressed common words ($\geq$ 6 characters, and 6< characters $\leq$ 12).

Method: Input data cards punched in 6 12-character fields, columns 1-72 are used. Each common word is left justified, right blank filled. End of data is indicated by three *** in the first three character positions of a field.

There are two lists:

1) Words of 6 characters or less
2) Words of 7-12 characters

Hash-addressing is by the division and remainder method without collision recovery. For double-word entries (7-12 characters) the two halves are added together (using ADLQ to inhibit overflow faults), and the total is then hashed. Tables have been made sufficiently large so as to reduce the probability of collision to an acceptable level.

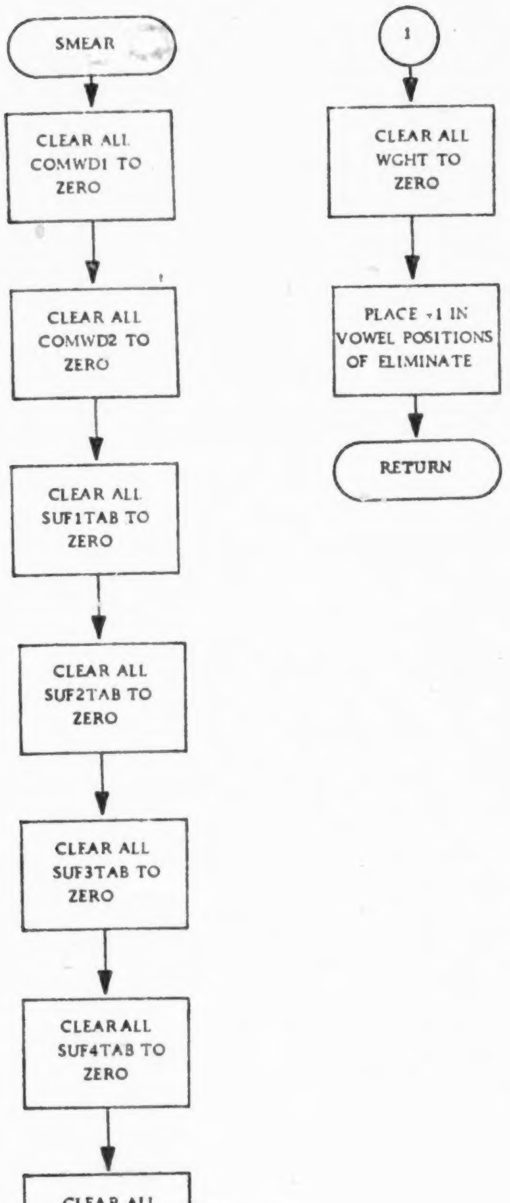

SMEAR
CLEAR ALL COMWD1 TO ZERO
CLEAR ALL COMWD2 TO ZERO
CLEAR ALL SUF1TAB TO ZERO
CLEAR ALL SUF2TAB TO ZERO
CLEAR ALL SUF3TAB TO ZERO
CLEARALL SUF4TAB TO ZERO
CLEAR ALL ROOTS TO ZERO
1
1
CLEAR ALL WGHT TO ZERO
PLACE +1 IN VOWEL POSITIONS OF ELIMINATE
RETURN

Figure V.10. SMEAR Flowchart

Figure V.11 is the complete flowchart for PLETAB. Figure V.12 illustrates the format of the punched cards for common words.

4. SUFTAB:

Function: To initialize the internal table of word suffixes for stem analysis during the run phase of micro-concordance generation.

Method: Suffixes are input from prepared punched cards or from pseudo-cards if the system is run from the permanent file and are grouped by ascending length from 1 to 4 characters. Each group is terminated by a field of three asterisks. Each group of suffixes is stored in the table corresponding to its length. Storage method is by the previously described division and remainer hashing system.

Operation:

1. The length counter is incremented. Eventually the counter will be used as an escape from the routine. That is, when length reaches 4, the routine will end.

2. Successive 6 bit bytes are taken from input cards until the break sequence, three asterisks, is encountered, when the length counter is checked. Each group of characters is zero filled on the right to conform to the method of table lookup used in the analysis section, then stored in the appropriate table depending on its length.

3. After each break sequence is encountered, a new table will be built, so that for suffixes of length one, there will be a suffix one table built for length two, a suffix two table, etc.

4. It would be possible to have additional tables of greater length simply by changing the length check to the appropriate maximum, and by providing additional tables for storage of the suffixes. Figure V.13 illustrates the flowchart for SUFTAB.

The four routines (PRERUN, SMEAR, PLETAB, and SUFTAB) finish up the set-up phase of the micro-concordance generation.

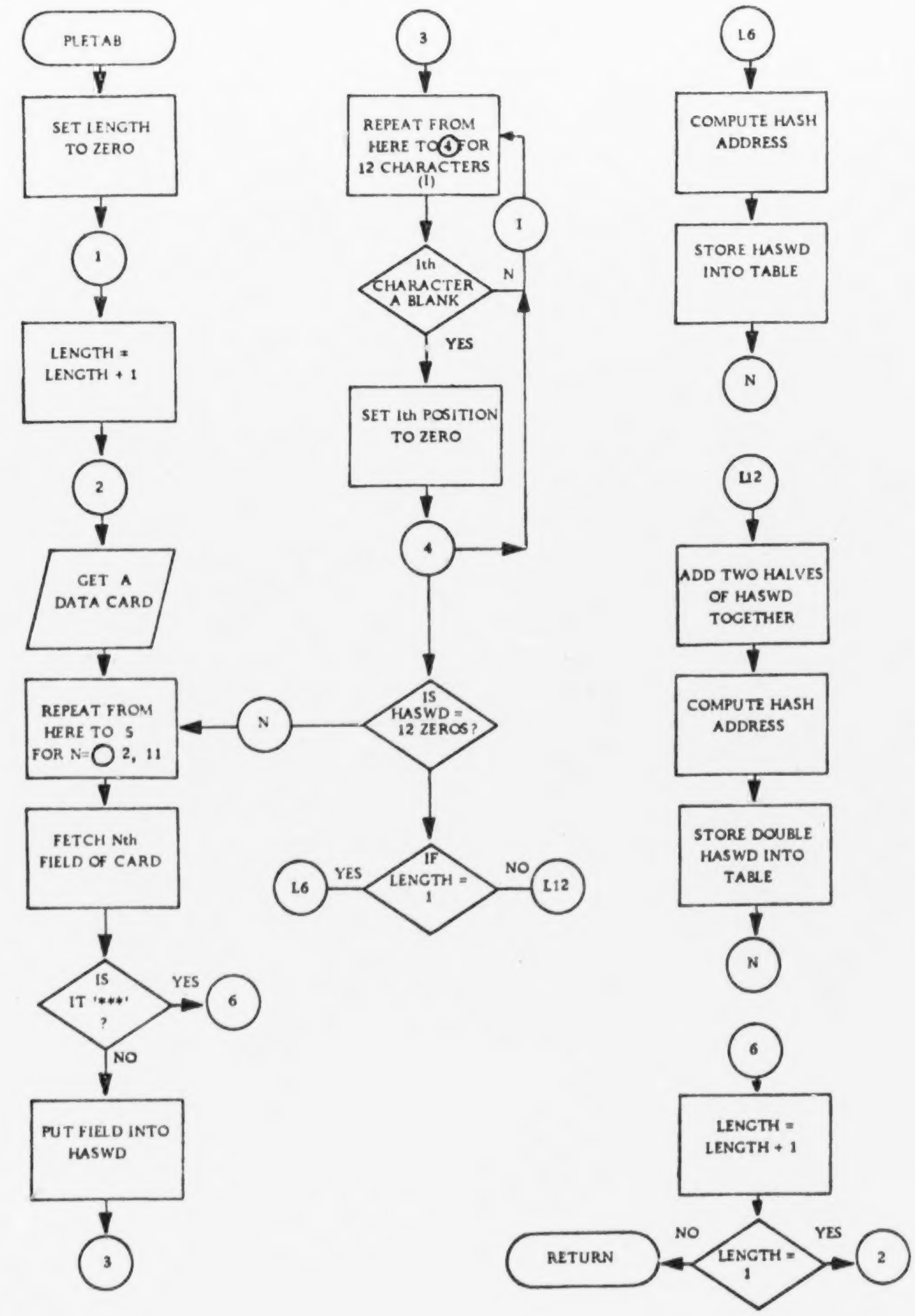
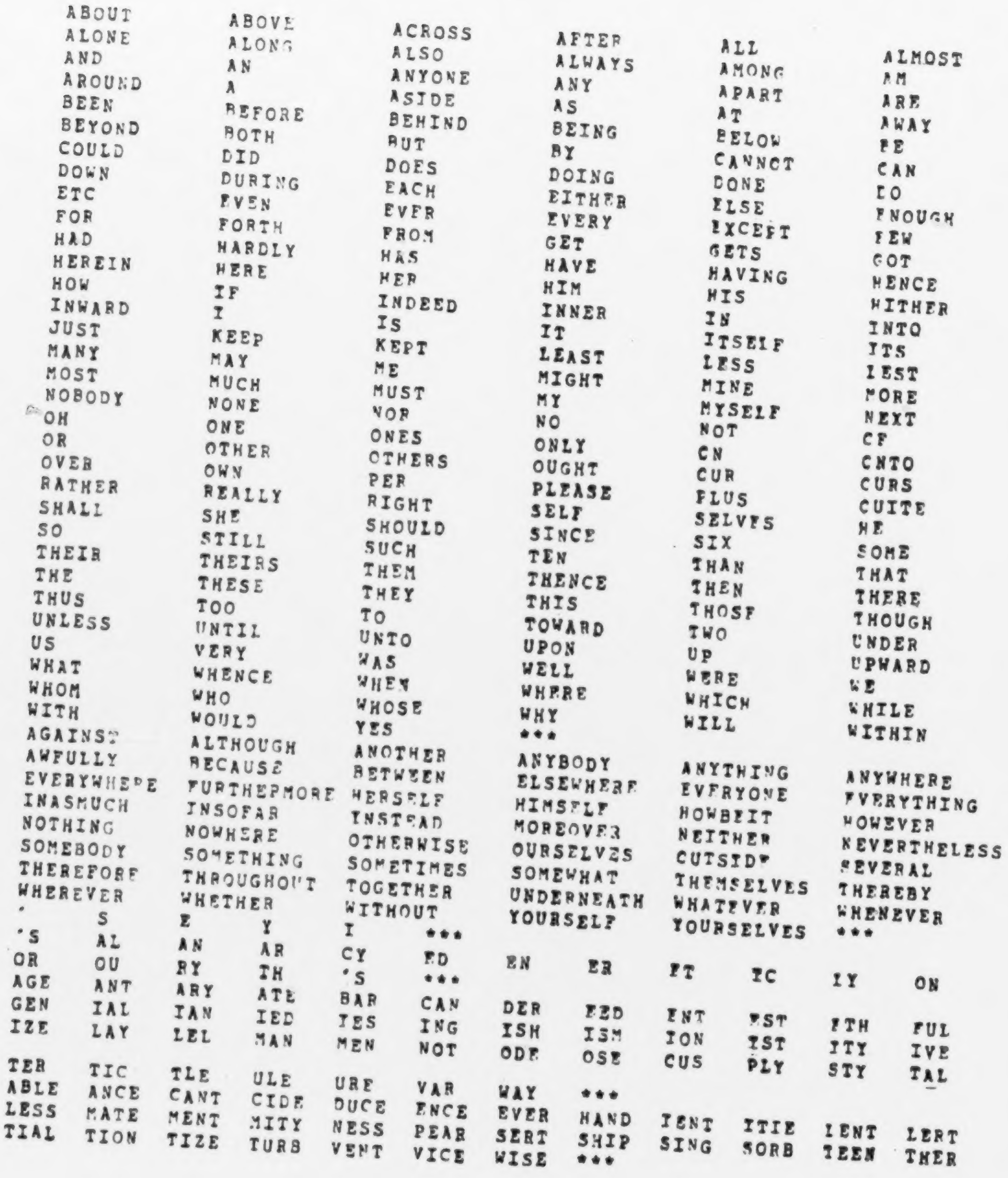

PLETAB
SET LENGTH TO ZERO
1
LENGTH = LENGTH + 1
2
GET A DATA CARD
REPEAT FROM HERE TO 5 FOR N=1, 2, 11
FETCH Nth FIELD OF CARD
IS IT '***' ?
YES
6
NO
PUT FIELD INTO HASWD
3
3
REPEAT FROM HERE TO 4 FOR 12 CHARACTERS (I)
Ith CHARACTER A BLANK
N
I
YES
SET Ith POSITION TO ZERO
4
IS HASWD = 12 ZEROS?
N
IF LENGTH = 1
YES
L6
NO
L12
L6
COMPUTE HASH ADDRESS
STORE HASWD INTO TABLE
N
L12
ADD TWO HALVES OF HASWD TOGETHER
COMPUTE HASH ADDRESS
STORE DOUBLE HASWD INTO TABLE
N
6
LENGTH = LENGTH + 1
LENGTH = 1
NO
RETURN
YES
2

Figure V.11. PLETAB Flowchart



```
ABOUT        ABOVE        ACROSS       AFTER        ALL          ALMOST
ALONE        ALONG        ALSO         ALWAYS       AMONG        AM
AND          AN           ANYONE       ANY          APART        ARE
AROUND       A            ASIDE        AS           AT           AWAY
BEEN         BEFORE       BEHIND       BEING        BELOW        BE
BEYOND       BOTH         BUT          BY           CANNOT       CAN
COULD        DID          DOES         DOING        DONE         DO
DOWN         DURING       EACH         EITHER       ELSE         ENOUGH
ETC          EVEN         EVER         EVERY        EXCEPT       FEW
FOR          FORTH        FROM         GET          GETS         GOT
HAD          HARDLY       HAS          HAVE         HAVING       HENCE
HEREIN       HERE         HER          HIM          HIS          HITHER
HOW          IF           INDEED       INNER        IN           INTO
INWARD       I            IS           IT           ITSELF       ITS
JUST         KEEP         KEPT         LEAST        LESS         LEST
MANY         MAY          ME           MIGHT        MINE         MORE
MOST         MUCH         MUST         MY           MYSELF       NEXT
NOBODY       NONE         NOR          NO           NOT          OF
OH           ONE          ONES         ONLY         ON           ONTO
OR           OTHER        OTHERS       OUGHT        OUR          OURS
OVER         OWN          PER          PLEASE       PLUS         QUITE
RATHER       REALLY       RIGHT        SELF         SELVES       RE
SHALL        SHE          SHOULD       SINCE        SIX          SOME
SO           STILL        SUCH         TEN          THAN         THAT
THEIR        THEIRS       THEM         THENCE       THEN         THERE
THE          THESE        THEY         THIS         THOSE        THOUGH
THUS         TOO          TO           TOWARD       TWO          UNDER
UNLESS       UNTIL        UNTO         UPON         UP           UPWARD
US           VERY         WAS          WELL         WERE         WE
WHAT         WHENCE       WHEN         WHERE        WHICH        WHILE
WHOM         WHO          WHOSE        WHY          WILL         WITHIN
WITH         WOULD        YES          ***
AGAINST      ALTHOUGH     ANOTHER      ANYBODY      ANYTHING     ANYWHERE
AWFULLY      BECAUSE      BETWEEN      ELSEWHERE    EVERYONE     EVERYTHING
EVERYWHERE   FURTHERMORE  HERSELF      HIMSELF      HOWBEIT      HOWEVER
INASMUCH     INSOFAR      INSTEAD      MOREOVER     NEITHER      NEVERTHELESS
NOTHING      NOWHERE      OTHERWISE    OURSELVES    OUTSIDE      SEVERAL
SOMEBODY     SOMETHING    SOMETIMES    SOMEWHAT     THEMSELVES   THEREBY
THEREFORE    THROUGHOUT   TOGETHER     UNDERNEATH   WHATEVER     WHENEVER
WHEREVER     WHETHER      WITHOUT      YOURSELF     YOURSELVES   ***
'     S     E     Y     I     ***
'S    AL    AN    AR    CY    ED
OR    OU    RY    TH    'S    ***   EN    ER    ET    IC    IY    ON
AGE   ANT   ARY   ATE   BAR   CAN   DER   EED   ENT   EST   ETH   FUL
GEN   IAL   IAN   IED   IES   ING   ISH   ISM   ION   IST   ITY   IVE
IZE   LAY   LEL   MAN   MEN   NOT   ODE   OSE   CUS   PLY   STY   TAL

TER   TIC   TLE   ULE   URE   VAR   WAY   ***
ABLE  ANCE  CANT  CIDE  DUCE  ENCE  EVER  HAND  IENT  ITIE  IENT  LERT
LESS  MATE  MENT  MITY  NESS  PEAR  SERT  SHIP  SING  SORB  TEEN  THER
TIAL  TION  TIZE  TURB  VENT  VICE  WISE  ***
```

Figure V.12. Punched Format of Common Words and Suffixes

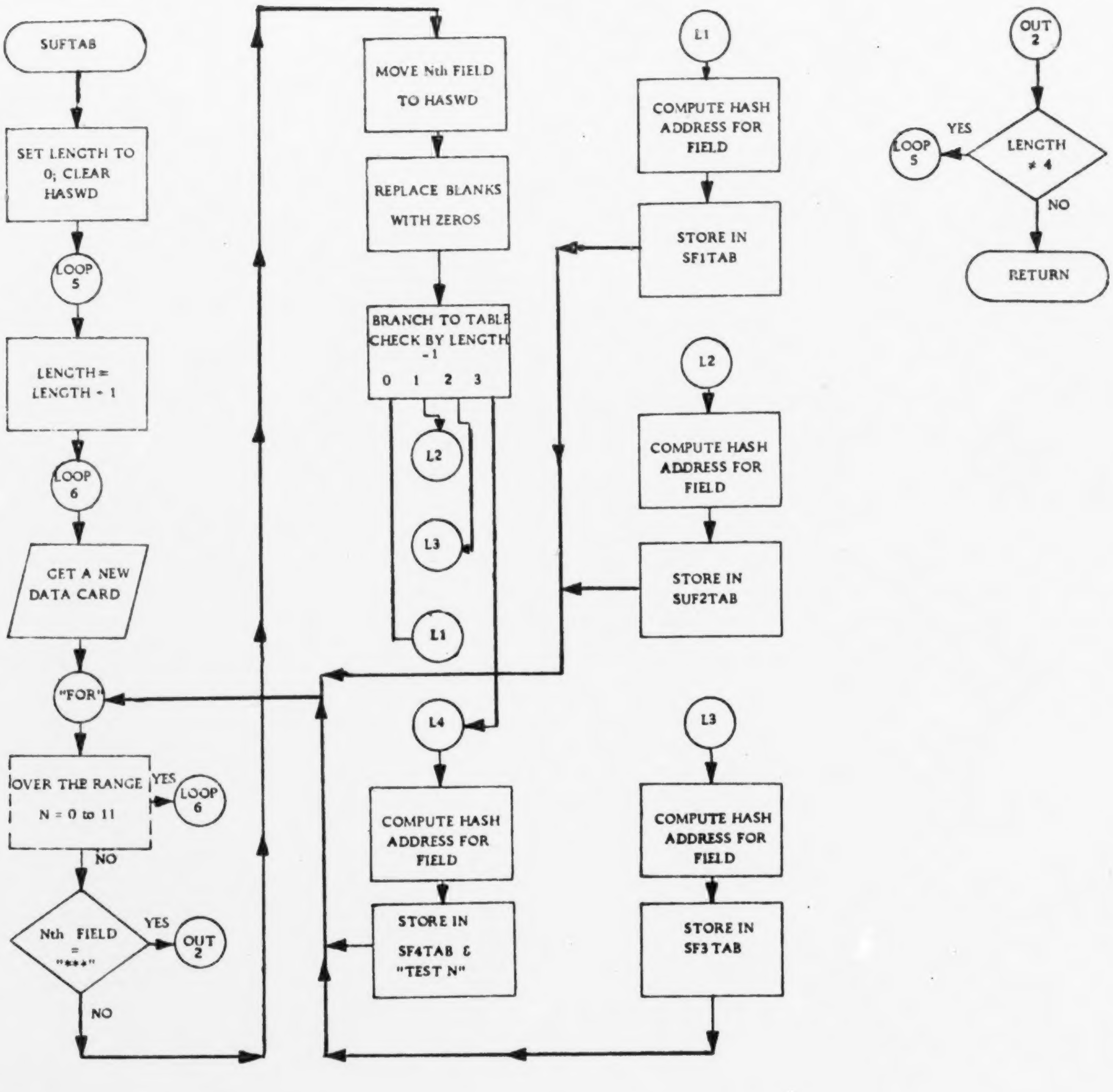

SUFTAB
SET LENGTH TO 0; CLEAR HASWD
LOOP 5
LENGTH = LENGTH - 1
LOOP 6
GET A NEW DATA CARD
"FOR"
OVER THE RANGE N = 0 to 11
YES
LOOP 6
NO
Nth FIELD = "***"
YES
OUT 2
NO
MOVE Nth FIELD TO HASWD
REPLACE BLANKS WITH ZEROS
BRANCH TO TABLE CHECK BY LENGTH -1
0 1 2 3
L2
L3
L1
L4
COMPUTE HASH ADDRESS FOR FIELD
STORE IN SF4TAB & "TEST N"
L1
COMPUTE HASH ADDRESS FOR FIELD
STORE IN SF1TAB
L2
COMPUTE HASH ADDRESS FOR FIELD
STORE IN SUF2TAB
L3
COMPUTE HASH ADDRESS FOR FIELD
STORE IN SF3 TAB
OUT 2
LENGTH ≠ 4
YES
LOOP 5
NO
RETURN

Figure V.13. SUFTAB Flowchart

Phase Two - (WORD)

The run phase of the micro-concordance generation consists of one large JOVIAL close routine called WORD. In this routine the entire text of a single abstract is read in, analyzed, and the output is built in a table to be put out in magnetic tape. WORD consists of essentially three sections, the first part of which reads in an entire abstract and breaks out the words that are contained in the abstract one at a time.

As each word is isolated, it is passed to the stem analysis routine, where it is then stemmed and entered into the table. However, before a word is passed to the stem analysis routine, its length is checked. If it is greater than 18 characters it is ignored. If it is greater than 12 characters, it cannot be a common word, and therefore the lookup in the common word table is ignored. If it is 12 or fewer characters in length and greater than one character, it then goes through the common word lookup which eliminates it if it falls in the list of common words. If the word which has been isolated is one character long, it is also ignored. Common word lookup is divided into two groups: those between 7 and 12 characters in length are looked up in the table of double word length common words, while those with 6 or less characters, but again greater than one character, are looked up in the common word table of one. The reason for this is the following: the hashing algorithm,(5) requires that input to it be one machine word. If we try to put in 2 machine words, we will end up ignoring one or the other half of the word which has been isolated. Therefore, some scheme is needed to combine the two. In our case the two halves of the isolated word, if it is 12 characters long, are added together. In order to avoid overflow faults on the computer, the add-logic instruction is used.

Once an isolated text word passes the common word lookup test it is then a candidate for stem analysis. During stem analysis the last n characters of an isolated word are looked up in the table starting with the table of 4 character suffixes first, by an algorithm described in RADC-TR-69-304, VI-10 to VI-13. In this program section the length of a word is gradually reduced, cutting off suffixes. The result of this is a stem or root, which is 12 or fewer characters in length, and is devoid of all the suffixes that we can find with the algorithm.

The stem is then passed down to the root routine which takes the truncated word, calculates a hash address (again the same algorithm), and stores it in the table of roots. First, however, a check is made to see if the root location is already occupied. If it is, it means we have an identical root and therefore no need to store the same root again. Instead, the corresponding counter for that root is incremented. The result of this is that at the end of a document, there is a hash generated table of roots and counts for each unique root in the entire abstract.

As soon as the last character in the abstract is processed (in our case this character "!"), the table of root and weights, which is what the counts are now called, are both packed. The reason for this is the following: the hash algorithm scatters entries in the roots and the weights tables fairly uniformly over the length of the table. In order to speed up the sort, which is the next logical step in the program, the zero elements must be squeezed out and the good elements pushed down toward the low end of the table, so that the sort can operate more quickly. The sorting algorithm is the modified version of the "sinking" or Shell sort(3).

When the root table has been sorted and the weight table items moved to the corresponding places, the entries in the roots and weights tables are then pulled out one at a time, put into an output table, and written out on magnetic tape. The routine then cycles back and gets more input data. If there are no more input data, that is, all abstracts are exhausted, the routine exits. The routine also prints the last accession number processed by the system because in our system of operation, the analysis or the concordance phase is broken up into several operational groups so that each input tape may be handled separately. As part of the concordance generation an additional tape is built called the LEADER tape. The reason for this is that the statistical filter routine requires the length of each concordance in the number of stems, that is, the number of running text words, in order to complete the statistical analysis that is written out for each concordance.

V.4.3 Operation of the Micro-Concordance Generator

In order to operate the micro-concordance generator, the following data sets are needed: a magnetic tape consisting of abstracts or any other text-type material, the terminating character for which is an "!". This is on magnetic tape in standard HISI 635 or 600 series tape, that is GEFRC standard format tape, and another magnetic tape for the output of the concordance. Another magnetic tape is used for the leaders. These are assigned the following logical file designators. The in-tape comes on logical file 02, output of the concordance is on logical file 01, and the leaders go out on logical file 03. Two additional files are used: file 05 is standard card input, and file 06 the printer on which the last used document number is printed. On the card reader file, or in the position that the cards that ordinarily occupy if the job is to be submitted by CARDIN, the tables must be arranged as in Figure V. 15. In other words, the first card must be a list of ignore characters, the second card a list of break characters (other than blank), followed by cards or pseudo-cards arranged in the format as illustrated. They are in the following order. The first group of data are the common words used in routine PLETAB to build the common word tables. After that the list of suffixes occurs, arranged in ascending number of characters, first the suffixes of one, then suffixes of two, then suffixes of three, and finally suffixes of four characters. Order of control cards and file cards as indicated in the Appendix to this technical note with other information about how to run the job. Figure V. 14 illustrates the complete flowchart for WORD.

V.4.4 Listing of Items, Tables, File Names and Other Pertinent Information

Tables

PEEKTB	PEEKTB is used to store the transfer vector for each character in the character set of the machine, so that during isolation of the text word from an abstract individual

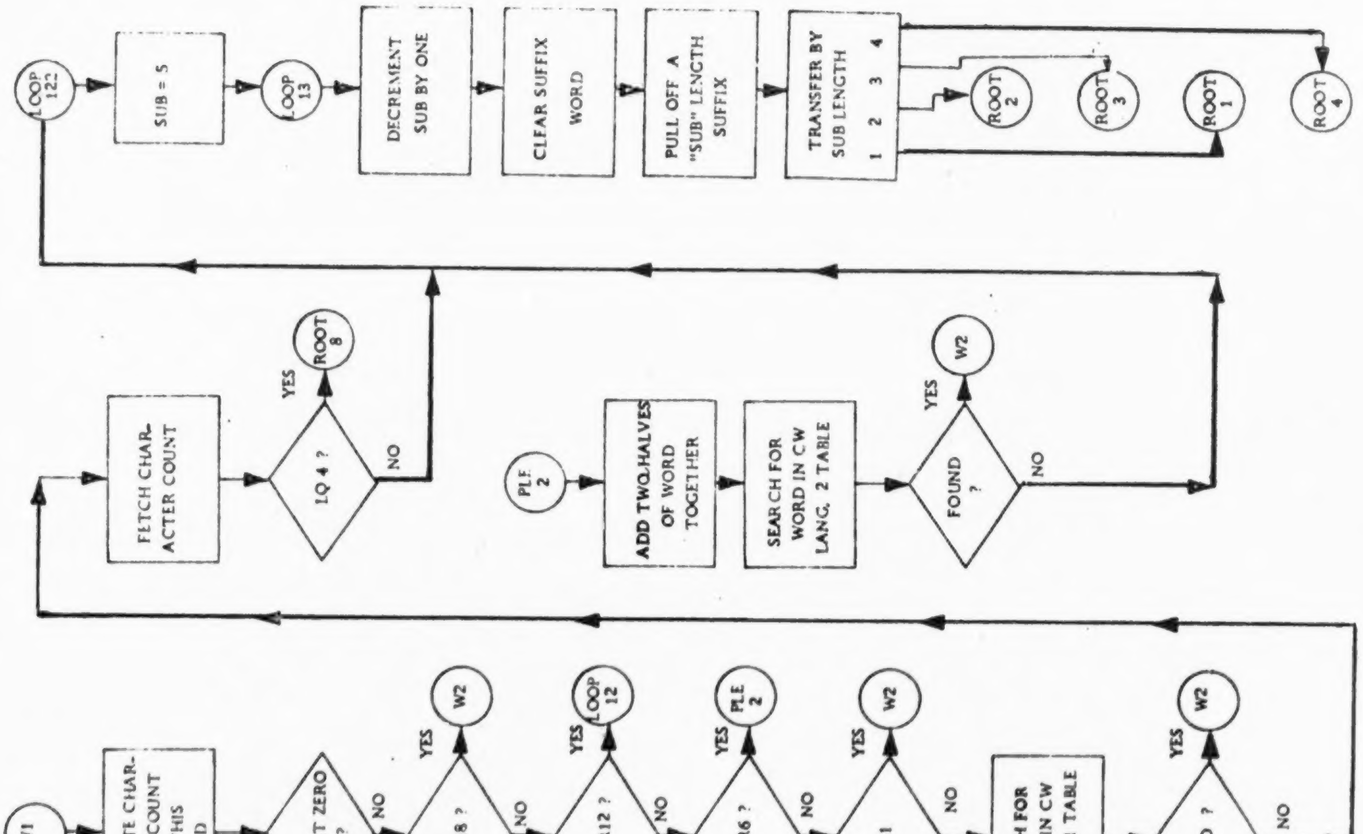

WORD
COUNT = 0
WORD
READ AN ABSTRACT
INCREMENT DOCNUM
END OF FILE ON INPUT ?
YES
NO
PRINT DOCNUM
RETURN
RESET TALLY COUNTERS
EW
FETCH NEXT CHARACTER
CHARACTER FETCH OK
YES
PKT
NO
PRO
MVB
STORE CHARACTER IN OUTPUT AREA
STORE OK ?
NO
W2
YES
W1
COMPUTE CHARACTER COUNT FOR THIS WORD
WAS IT ZERO ?
YES
NO
GR18 ?
YES
W2
NO
GR12 ?
YES
LOOP 12
NO
GR6 ?
YES
PLE 2
NO
EQ 1
YES
W2
NO
SEARCH FOR WORD IN CW LENGTH 1 TABLE
FOUND ?
YES
W2
NO
FETCH CHARACTER COUNT
LQ 4 ?
YES
ROOT 8
NO
PLE 2
ADD TWO HALVES OF WORD TOGETHER
SEARCH FOR WORD IN CW LANG, 2 TABLE
FOUND ?
YES
W2
NO
LOOP 122
SUB = 5
LOOP 13
DECREMENT SUB BY ONE
CLEAR SUFFIX WORD
PULL OFF A "SUB" LENGTH SUFFIX
TRANSFER BY SUB LENGTH
1 2 3 4
ROOT 2
ROOT 3
ROOT 1
ROOT 4

Figure V.14. WORD Flowchart

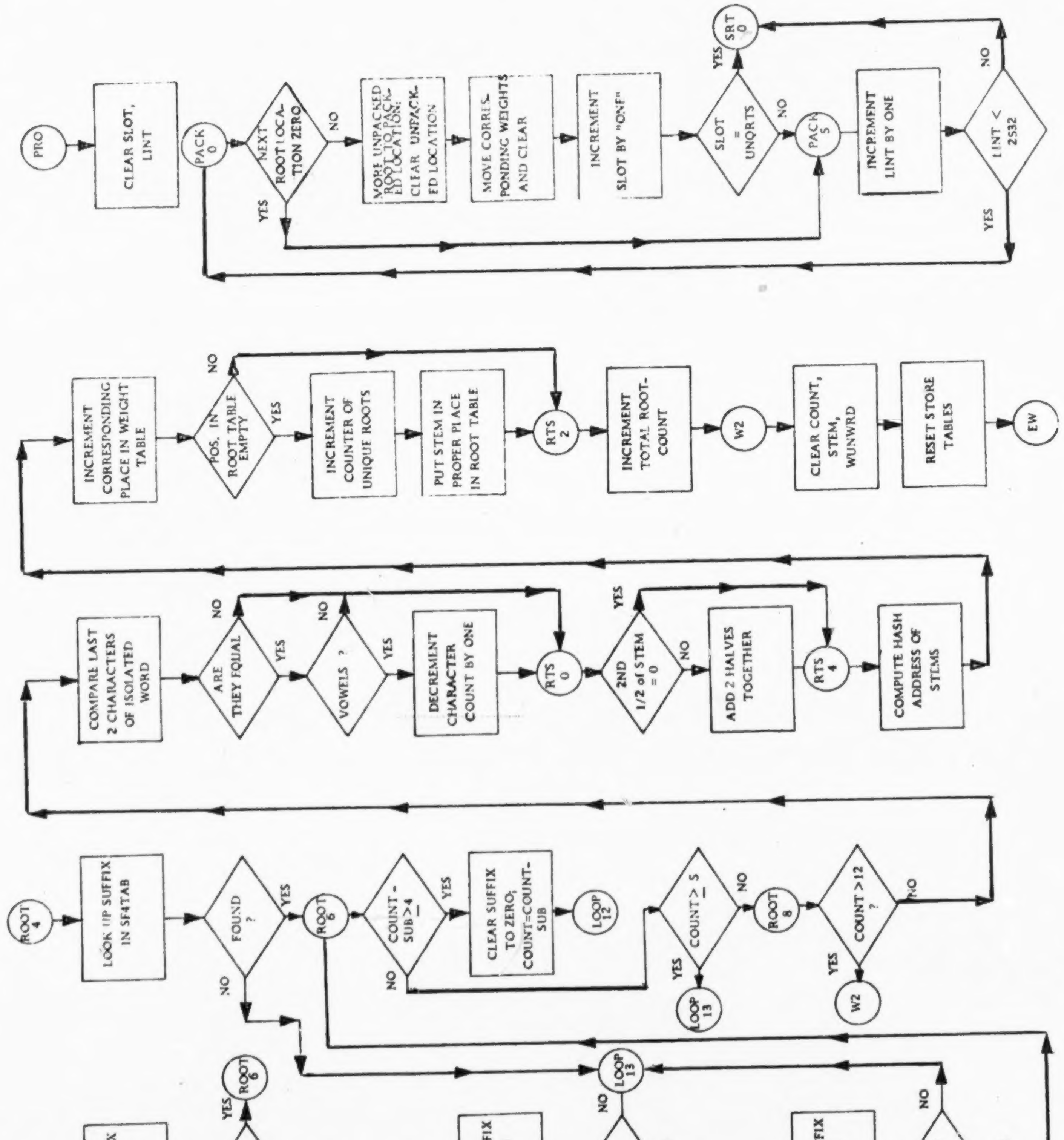

ROOT 1
LOOK UP SUFFIX IN SF1TAB
FOUND ?
YES
ROOT 6
NO
ROOT 8
ROOT 2
LOOK UP SUFFIX IN SF2TAB
FOUND ?
YES
ROOT 6
NO
LOOP 13
ROOT 3
LOOK UP SUFFIX IN SF3TAB
FOUND ?
YES
NO
ROOT 4
LOOK UP SUFFIX IN SF4TAB
FOUND ?
YES
NO
ROOT 6
COUNT - SUB ≥ 4
YES
NO
CLEAR SUFFIX TO ZERO; COUNT=COUNT-SUB
LOOP 12
COUNT ≥ 5
YES
LOOP 13
NO
ROOT 8
COUNT > 12
YES
W2
NO
COMPARE LAST 2 CHARACTERS OF ISOLATED WORD
ARE THEY EQUAL
NO
YES
VOWELS ?
NO
YES
DECREMENT CHARACTER COUNT BY ONE
RTS 0
2ND 1/2 of STEM = 0
YES
NO
ADD 2 HALVES TOGETHER
RTS 4
COMPUTE HASH ADDRESS OF STEMS
INCREMENT CORRESPONDING PLACE IN WEIGHT TABLE
POS. IN ROOT TABLE EMPTY
NO
YES
INCREMENT COUNTER OF UNIQUE ROOTS
PUT STEM IN PROPER PLACE IN ROOT TABLE
RTS 2
INCREMENT TOTAL ROOT-COUNT
W2
CLEAR COUNT, STEM, WUNWRD
RESET STORE TABLES
EW
PRO
CLEAR SLOT, LINT
PACK 0
NEXT ROOT LOCATION ZERO
YES
NO
MOVE UNPACKED ROOT TO PACKED LOCATION; CLEAR UNPACKED LOCATION
MOVE CORRESPONDING WEIGHTS AND CLEAR
INCREMENT SLOT BY "ONE"
SLOT = UNQRTS
YES
SRT 0
NO
PACK 5
INCREMENT LINT BY ONE
LINT < 2532
NO
YES

Figure V.14. (Continued)

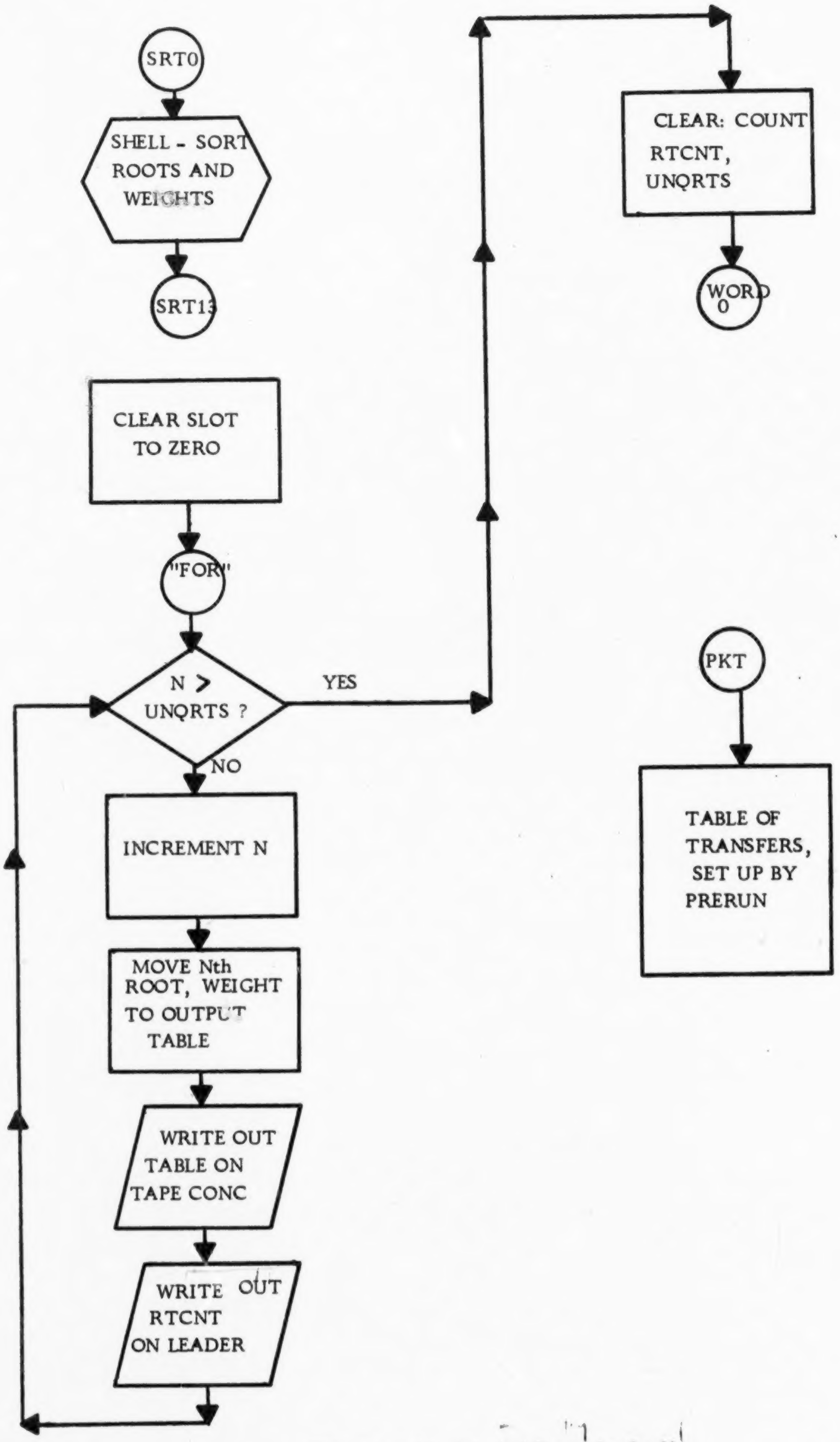

SRT0
SHELL - SORT ROOTS AND WEIGHTS
SRT13
CLEAR SLOT TO ZERO
"FOR"
N > UNQRTS ?
YES
NO
INCREMENT N
MOVE Nth ROOT, WEIGHT TO OUTPUT TABLE
WRITE OUT TABLE ON TAPE CONC
WRITE OUT RTCNT ON LEADER
CLEAR: COUNT RTCNT, UNQRTS
WORD 0
PKT
TABLE OF TRANSFERS, SET UP BY PRERUN

Figure V.14. (Concluded)

are checked for one to four status conditions. They can either be legal characters, characters which indicate an end of a word called break characters, or characters which will be ignored, such as garbage characters; or an escape character, which in our character set is an exclamation point.

COMWD1 & 2	COMWD1 & 2 store the tables of common words, separated into common words of one machine word, and common words of two machine words, in length.
SUF1TB SUF2TB SUF3TB SUF4TB	SUFTB's 1-4 store the suffixes which will be checked in stem analysis, grouped by their length. (Stem suffixes of length character one character two characters three characters four characters).
ROOTS WGHT	The tables ROOTS and WGHT store the actual concordance information. In table ROOTS are hash stored, the stems which have been analyzed out of the text; the table WGHT stores the corresponding weight or the number of times that stem occurs in the text. The method of addressing is the same for both, and they occupy corresponding places in each table.
CARD	Used for storing input cards.
ABSTOR	The entire text of one abstract is read into ABSTOR for analysis.
ELIM	Table ELIM is used during stem analysis to hold the transfer vector for vowel checking in order to eliminate dual vowels.
CONCRD	Table CONCRD is a dummy four machine word area into which the information about each root and weight input just prior to putting it out on magnetic tape. This table is changed constantly and does not have any permanent contents.

Items

WUNWRD	This is the area into which isolated words are placed prior to stem analysis.
RTCNT	This cell holds the running total count of stems isolated in the text. This differs from the next item, UNQRTS, which holds only the count of the unique roots. RTCNT is used for putting out the LEADER tape, used by the statistical filter.

HASWD	This is a holding area used for computing the hash address during the building of the tables, in the pre-run phase of the job.
LENGTH	Also used to hold the length of the number of characters of fields during the pre-run phase.
COUNT	Same as the above.
CW1LEN CW2LEN	These are pre-set constants to indicate the number of machine words in common word 1 table, common word 2 table.
ST1LEN ST2LEN ST3LEN ST4LEN	These also indicate the number of machine words in the suffix table of lengths 1-4, used during hash address generation.
W1	A temporary cell.
LINT	Used during table packing.
STEM	Used during stem analysis.
SUFFIX	Used during stem analysis.
SLOT	Used during table packing.
SUB	Used during stem analysis.
NXBYTE	Also used during stem analysis.
DOCNUM	Used to store the number of the document being analyzed.
EMPTY	A pre-set value of zero.
II JJ KK DD LL W2	Used as pointers during the sinking or SHELL sort.
TALLIES TALLY1 TALLY2 TALLY3 TALLY4 TALLY5 TALLY6	All used for Sequence Character indirect addressing during fetching and storing of characters.

TRATST	The value for transferring to ENDWRD in the text.
TRAWD1	To transfer to WORD1.
TRAMOV	To transfer to MOVEBY.
ENDOC	To PROCES.
	These 4 cells are used during the set-up of PEEKTB, during the pre-run phase, and they are transferred to different places in the program which are needed depending on the various status of different characters.
Files	There are 5 tape or other type of files which must be defined for the operation of the program.
REDER	The standard card input unit.
PRNT	Used for standard output printer.
CONC	Used for concordance output.
INTAPE	Used for concordance input.
LEADER	Used for leader output to the statistical filter.

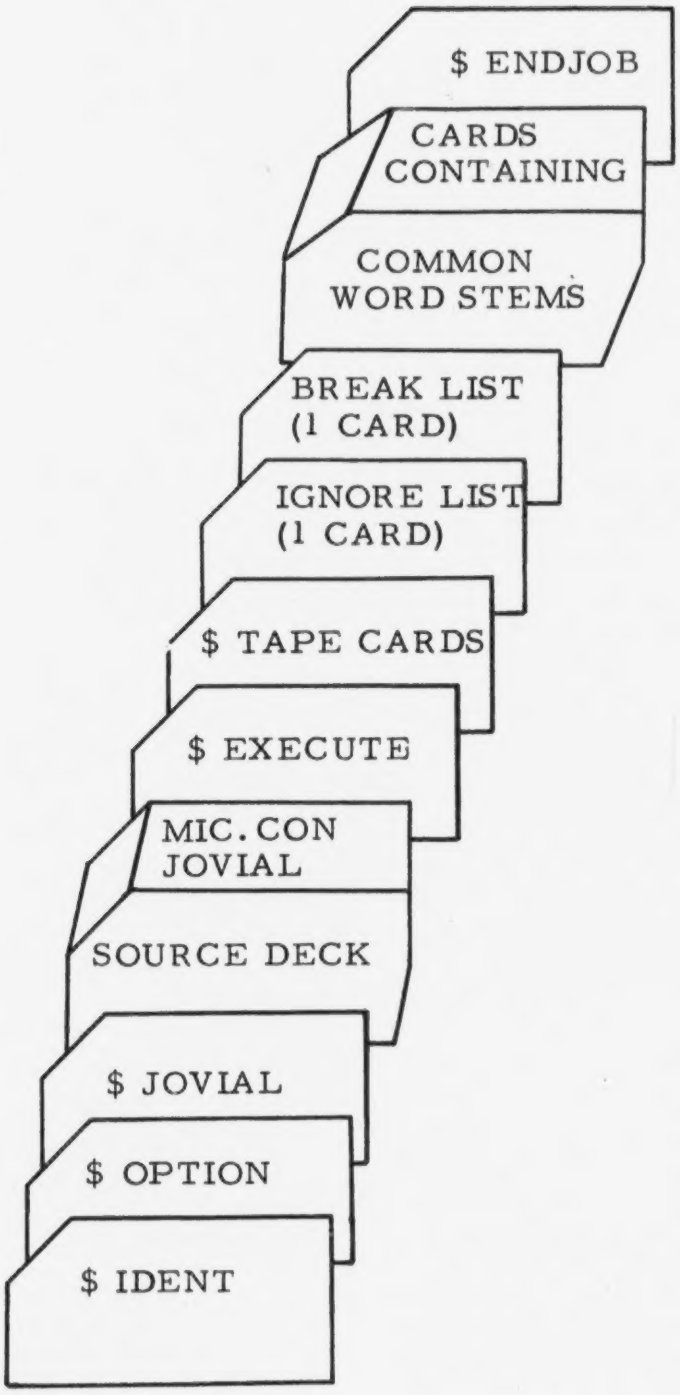

$ ENDJOB
CARDS CONTAINING
COMMON WORD STEMS
BREAK LIST (1 CARD)
IGNORE LIST (1 CARD)
$ TAPE CARDS
$ EXECUTE
MIC.CON JOVIAL
SOURCE DECK
$ JOVIAL
$ OPTION
$ IDENT

Figure V.15. Deck Set-Up for MIC.CON Run

```
1321T 01  10-09-70   13.463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM MAIN
  0000       START                                                         00000200          0001
  0001            TABLE PEEKTB R 64 P N $                                  00000300          0002
  0001             BEGIN ''PEEKTB''                                        00000400          0003
  0002              ITEM CHARACTER I 36 S $                                00000500          0004
  0003             END ''PEEKTB''                                          00000600          0005
  0003            TABLE COMWD1 R 4333 S $                                  00000700          0006
  0003             BEGIN ''COMWD1''                                        00000800          0007
  0004              ITEM COMMON1 H 6 $                                     00000900          0008
  0005             END   ''COMWD1''                                        00001000          0009
  0005            TABLE COMWD2 R 1533 P 2 $                                00001100          0010
  0005             BEGIN ''COMWD2''                                        00001200          0011
  0006              ITEM COMMON2 H 12 0 0 $                                00001300          0012
  0007             END ''COMWD2''                                          00001400          0013
  0007            TABLE SUF1TB R 25 P $                                    00001500          0014
  0007             BEGIN ''SUF1TB''                                        00001600          0015
  0008              ITEM SUF1 H 6 $                                        00001700          0016
  0009             END   ''SUF1TB''                                        00001800          0017
  0009            TABLE SUF2TB R 275 P $                                   00001900          0018
  0009             BEGIN ''SUF2TB''                                        00002000          0019
  0010              ITEM SUF2 H 6 $                                        00002100          0020
  0011             END   ''SUF2TB''                                        00002200          0021
  0011            TABLE SUF3TB R 501 P $                                   00002300          0022
  0011             BEGIN ''SUF3TB''                                        00002400          0023
  0012              ITEM SUF3 H 6 $                                        00002500          0024
  0013             END   ''SUF3TB''                                        00002600          0025
  0013            TABLE SUF4TB R 999 P $                                   00002700          0026
  0013             BEGIN ''SUF4TB''                                        00002800          0027
  0014              ITEM SUF4 H 6 $                                        00002900          0028
  0015             END   ''SUF4TB''                                        00003000          0029
  0015            TABLE ROOTS R 2533 S 2 $                                 00003100          0030
  0015             BEGIN ''ROOTS''                                         00003200          0031
  0016              ITEM ROOT H 12 0 0 $                                   00003300          0032
  0017             END ''ROOTS''                                           00003400          0033
  0017            TABLE WGHT R 2533 S N $                                  00003500          0034
  0017             BEGIN ''WGHT''                                          00003600          0035
  0018              ITEM WEIGHT I 36 S $                                   00003700          0036
  0019             END ''WGHT''                                            00003800          0037
  0019            TABLE CARD R 14 S 1 $                                    00003900          0038
  0019             BEGIN '' CARD''                                         00004000          0039
  0020              I EM XCARD H 6 0 0 $                                   00004100          0040
  0021             EP  ''CARD''                                            00004200          0041
  0021            TAP'E ABSTOR R 500 S 1 $                                 00004300          0042
  0021             BEGIN ''ABSTORE''                                       00004400          0043
  0022              ITEM ABSTOR'WRD H 6 0 0 $                              00004500          0044
  0023             END ''ABSTORE''                                         00004600          0045
  0023            TABLE ELIM R 64 S 1 $                                    00004640          0046
  0023             BEGIN ''ELIM''                                          00004645          0047
  0024              ITEM ELMCHR H 6 0 0 $                                  00004650          0048
  0025             END ''ELIM''                                            00004655          0049
```

Figure V.15a. MIC.CON Listing

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0025              TABLE CONCRD R 4 S N $                           00004660      0050
 0025               BEGIN ''N''                                     00004665      0051
 0026                ITEM CONWRD H 6 $                              00004670      0052
 0027               END ''N''                                       00004675      0053
 0027              ITEM WUNWRD H 60 $                               00004900      0054
 0028              ITEM RTCNT I 36 S P 0 $                          00005000      0055
 0029              ITEM UNORTS I 36 S P 0 $                         00005100      0056
 0030              ITEM HASWD H 12 P 12H(000000000000) $            00005200      0057
 0031              ITEM LENGTH I 35 U P 0 $                         00005500      0058
 0032              ITEM COUNT I 35 U P 0 $                          00005600      0059
 0033              ITEM FLAG I 35 U P 0 $                           00005700      0060
 0034              ITEM CW1LEN I 35 U P 4333 $                      00005800      0061
 0035              ITEM CW2LEN I 35 U P 1533 $                      00005900      0062
 0036              ITEM     JT1LEN I 35 U P 25 $                    00006000      0063
 0037              ITEM ST2LEN I 35 U P 275 $                       00006100      0064
 0038              ITEM ST3LEN I 35 U P 501 $                       00006200      0065
 0039              ITEM ST4LEN I 35 U P 999 $                       00006300      0066
 0040              ITEM RTLEN I 35 U P 2533 $                       00006400      0067
 0041              ITEM     W1 I 36 S $                             00006500      0068
 0042              ITEM INT I 36 S $                                00006600      0069
 0043              ITEM LINT I 36 S P 0 $                           00006610      0070
 0044              ITEM FIND I 35 U $                               00006700      0071
 0045              ITEM STEM H 12 P 12H(000000000000) $             00006800      0072
 0046              ITEM SUFFIX H 6 P 6H(000000) $                   00006900      0073
 0047              ITEM RELADR I 35 U $                             00007000      0074
 0048              ITEM SLOT I 35 U P 0 $                           00007100      0075
 0049              ITEM MASK H 1 P O(77) $                          00007200      0076
 0050              ITEM ABWRD H 6 $                                 00007300      0077
 0051              ITEM SUB I 36 S P 0 $                            00007400      0078
 0052              ITEM NXBYTE H 1 $                                00007500      0079
 0053              ITEM TEMP I 36 S $                               00007510      0080
 0054              ITEM DOCNUM I 36 S P 0 $                         00007520      0081
 0055              ITEM EMPTY I 36 S P 0 $                          00007550      0082
 0056              ITEM     II I 36 S $                             00007600      0083
 0057              ITEM     JJ I 36 S $                             00007700      0084
 0058              ITEM     KK I 36 S $                             00007800      0085
 0059              ITEM     DD I 36 S $                             00007900      0086
 0060              ITEM     LL I 36 S $                             00008000      0087
 0061              ITEM     W2 H 12 $                               00008100      0088
 0062              DIRECT                                           00008200      0089
 0063        TALLY1 ZERO     CARD+1                                 00008300      0090
 0064        TALLY2 ZERO     CARD+1                                 00008400      0091
 0065        TALLY3 ZERO     ABSTOR+1                               00008500      0092
 0066        TALLY4 ZERO     ABSTOR+1         TALLY COUNTER         00008600      0093
 0067        TALLY5 ZERO     WUNWRD                                 00008700      0094
 0068        TALLY6 ZERO     WUNWRD                                 00008800      0095
 0069        TRATST TRA      ENDWRD                                 00008900      0096
 0070        TRAWD1 TRA      WORD1                                  00009000      0097
 0071        TRAMOV TRA      MOVEBY                                 00009100      0098
 0072        ENDOC  TRA      PROCES                                 00009200      0099
```

Figure V.15a. (Continued)

```
1321T 01  10=09=70   13.463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

0073           JOVIAL                                                               00009300          0100
0074           FILE REDER V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R05 $         00009400          0101
0075           FILE PRNT V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R06 $          00009500          0102
0076           FILE CONC V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R01 $          00009600          0103
0077           FILE INTAPE V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R02 $        00009700          0104
0078           FILE LEADER V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R03 $        00009710          0105
0079           SWITCH MODE1 = (L6,L12) $                                            00009800          0106
0080           SWITCH MODE2 = (L1,L2,L3,L4) $                                       00009900          0107
0081           SWITCH MODE8=(WORD2,LOOP12) $                                        00010200          0108
0082           SWITCH MODE10 = (ROOT1,ROOT2,ROOT3,ROOT4) $                          00010300          0109
0083    GO.    IN(1,INTAPE) $  ''PROVIDE A STARTING POINT FOR JOVIAL''              00010400          0110
0084           IN(1,REDER) $                                                        00010500          0111
0085           OUT(1,PRNT) $                                                        00010600          0112
0086           OUT(1,CONC) $                                                        00010610          0113
0087           OUT(1,LEADER) $                                                      00010620          0114
0087    ..     THIS IS SET-UP CODE FOR TALLY CELLS ''                               00010700          0115
0088           DIRECT                                                               00010800          0116
0089           LDA      60,DL              CODE TO SET UP                           00010900          0117
0090           ALR      6                  SC-MODIFIED                              00011000          0118
0091           STCA     TALLY5,06          TALLY COUNTERS                           00011100          0119
0092           STCA     TALLY6,06                                                   00011200          0120
0093           LDA      65,DL              FIRST VALUE                              00011300          0121
0094           ALR      6                                                           00011400          0122
0095           STCA     TALLY1,06                                                   00011500          0123
0096           STCA     TALLY2,06                                                   00011600          0124
0097           LDA      3000,DL            SECOND VALUE                             00011700          0125
0098           ALR      6                                                           00011800          0126
0099           STCA     TALLY3,06                                                   00011900          0127
0100           STCA     TALLY4,06                                                   00012000          0128
0101           JOVIAL                                                               00012100          0129
0102             GOTO PRERUN $                                                      00012200          0130
0103             GOTO SMEAR $                                                       00012300          0131
0104             GOTO PLETAB $                                                      00012400          0132
0105             GOTO SUFTAB $                                                      00012500          0133
0106             GOTO WORD $                                                        00012600          0134
0107    FINI,  IN(2,REDER) $                                                        00012800          0135
0108           IN(2,INTAPE) $                                                       00013000          0136
0109           OUT(2,PRNT) $                                                        00013010          0137
0110           OUT(2,CONC) $                                                        00013100          0138
0111           OUT(2,LEADER) $                                                      00013110          0139
0112    STOP $                                                                      00013200          0140
0112    ''*************************************************************''          00013300          0141
0112    ..                          PRERUN                             ..          00013400          0142
0112    ''*************************************************************''          00013500          0143
0114           CLOSE PRERUN $ BEGIN                                                 00013600          0144
0114    ..             INITIALIZE TABLE TO BRANCH FOR ALL VALID CHARACTERS''        00013700          0145
0115           FOR I = ALL(PEEKTB) $                                                00013800          0146
0116           BEGIN                                                                00013900          0147
0117           DIRECT                                                               00014000          0148
0118           LDA      TRAMOV                                                      00014100          0149
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

0119          STA     PEEKTB+1,2                                        00014200
0120          JOVIAL                                                    00014300
0121          END                                                       00014400
0121    ''    SET BLANK AS DEFAULT BREAK CHARACTER ''                   00014500
0122          DIRECT                                                    00014600
0123          LDA     TRAWD1                                            00014700
0124          STA     PEEKTB+17                                         00014800
0125          JOVIAL                                                    00014900
0125    ''    NOW READ LIST OF IGNORE CHARACTERSS ONE TC A COLUMN''     00015000
0126    IGN.  IO999(0,REDER,0,CARD,14) $                                00015100
0127          WAIT(IO999) $                                             00015200
0128          IF REDER EQ V(EOF) $                                      00015400
0130          IO090(3,PRNT,0,6H(ABORT1),1) $ STOP $                     00015500
0130    ''            AN EOF IS ILLEGAL HERE ''                         00015600
0131          DIRECT                                                    00015700
0132          LDQ     TRATST          SET Q W/ BRANCH INSTRUCTION       00015800
0133    IGNS  LDA     TALLY1,SC       FETCH ONE CHARACTER TO CLEARED A  00015900
0134          CMPA    16,DL           CHECK FOR A BLANK CHARACTER       00016000
0135          TZE     *+3             IF BLANK,PROCESS BREAK CHARS      00016100
0136          STQ     PEEKTB+1,AL     SET TABLE ENTRY                   00016200
0137          TTF     IGNS            BRANCH FOR MORE                   00016300
0138          JOVIAL                                                    00016400
0139          IO998(0,REDER,0,CARD,14) $                                00016500
0140          WAIT(IO998) $                                             00016600
0141          IF REDER EQ V(EOF) $                                      00016700
0143          IO091(3,PRNT,0,6H(ABORT2),1) $ STOP $                     00016800
0143    ''    AN EOF IS ILLEGAL HERE TOO ''                             00016900
0144          DIRECT                                                    00017000
0145          LDQ     TRAWD1          SET UP Q W/ BRANCH INSTRUCTION    00017100
0146    BRKS  LDA     TALLY2,SC       FETCH ONE CHARACTER               00017200
0147          CMPA    16,DL           CHECK FOR A BLANK CHARACTER       00017300
0148          TZE     *+3             IF BLANK,DONE W/PRERUN            00017400
0149          STQ     PEEKTB+1,AL     SET TABLE ENTRY                   00017500
0150          TTF     BRKS            BRANCH FOR MORE                   00017600
0151          LDA     ENDOC                                             00017700
0152          STA     PEEKTB+64                                         00017800
0153          JOVIAL                                                    00017900
0154          RETURN $                                                  00018000
0155                  END '' PRERUN PHASE ''                            00018100
0155    ''*************************************************************'' 00018200
0155    ''                            SMEAR                            '' 00018300
0155    ''*************************************************************'' 00018400
0156            CLOSE SMEAR $                                           00018500
0157             BEGIN ''SMEAR''                                        00018600
0158              FOR N=ALL(COMWD1)  $                                  00018700
0159               BEGIN ''N''                                          00018800
0160              COMMON1($N$)=6H(000000) $                             00018900
0161               END ''N''                                            00019000
0162             FOR N=ALL(COMWD2) $                                    00019100
0163              BEGIN ''N''                                           00019200
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

0164                     COMMON2($N$)=12H(000000000000) $                               00019300          0200
0165                    END ''N''                                                       00019400          0201
0166                   FOR N=ALL(SUF1TB) $                                              00019500          0202
0167                    BEGIN ''N''                                                     00019600          0203
0168                     SUF1($N$)=6H(000000) $                                         00019700          0204
0169                    END ''N''                                                       00019800          0205
0170                   FOR N=ALL(SUF2TB) $                                              00019900          0206
0171                    BEGIN ''N''                                                     00020000          0207
0172                     SUF2($N$)=6H(000000) $                                         00020100          0208
0173                    END ''N''                                                       00020200          0209
0174                   FOR N=ALL(SUF3TB) $                                              00020300          0210
0175                    BEGIN ''N''                                                     00020400          0211
0176                     SUF3($N$)=6H(000000) $                                         00020500          0212
0177                    END ''N''                                                       00020600          0213
0178                   FOR N=ALL(SUF4TB) $                                              00020700          0214
0179                    BEGIN ''N''                                                     00020800          0215
0180                     SUF4($N$)=6H(000000) $                                         00020900          0216
0181                    END ''N''                                                       00021000          0217
0182                   FOR N=ALL(ROOTS) $                                               00021100          0218
0183                    BEGIN ''N''                                                     00021200          0219
0184                     ROOT($N$)=12H(000000000000) $                                  00021300          0220
0185                    END ''N''                                                       00021400          0221
0186                   FOR N=ALL(WGHT) $                                                00021500          0222
0187                    BEGIN ''N''                                                     00021600          0223
0188                     WEIGHT($N$)=0 $                                                00021700          0224
0189                    END ''N''                                                       00021800          0225
0190                   FOR N=ALL(ELIM) $                                                00021810          0226
0191                    BEGIN ''N''                                                     00021820          0227
0192                     ELMCHR($N$)=6H(000000) $                                       00021830          0228
0193                    END ''N''                                                       00021840          0229
0194                   ELMCHR($17$)=6H(000001) $                                        00021850          0230
0195                   ELMCHR($21$)=6H(000001) $                                        00021855          0231
0196                   ELMCHR($25$)=6H(000001) $                                        00021860          0232
0197                   ELMCHR($38$)=6H(000001) $                                        00021865          0233
0198                   ELMCHR($52$)=6H(000001) $                                        00021870          0234
0199                    END ''SMEAR''                                                   00021900          0235
0199          ''*******************************************************************''   00022000          0236
0199          ''                      PLETAB                                       ''   00022100          0237
0199          ''*******************************************************************''   00022200          0238
0200                 CLOSE PLETAB $                                                     00022300          0239
0201                  BEGIN ''PLETAB''                                                  00022400          0240
0202                   LENGTH=0 $                                                       00022500          0241
0203          LOOP1.   LENGTH=LENGTH+1 $                                                00022600          0242
0204          LOOP2.   GETCRD(=CARD) $                                                  00022700          0243
0205                   FOR N=0,2,11 $                                                   00022800          0244
0206                    BEGIN ''N''                                                     00022900          0245
0207                     IF XCARD($N$) EQ O(545454202020) $                             00023000          0246
0208                     GOTO OUT1 $                                                    00023100          0247
0209                     BYTE($0,6$)(HASWD)=XCARD($N$) $                                00023200          0248
0210                     BYTE($6,6$)(HASWD)=XCARD($N+1$) $                              00023300          0249
```

Figure V.15a. (Continued)

```
1321I 01  10=09=70   13.463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46
 0211             FOR I=0,1,11 $                                              00023400        0250
 0212              BEGIN ''I''                                                00023500        0251
 0213               IF BYTE($I,1$)(HASWD) NQ 1H( ) $                          00023600        0252
 0214             TEST $                                                      00023700        0253
 0215             BYTE($I,1$)(HASWD)=1H(Q) $                                  00023800        0254
 0216              END ''I''                                                  00023900        0255
 0217             IF HASWD EQ 12H(000000000000) $                             00024000        0256
 0218             TEST N $                                                    00024100        0257
 0219             GOTO MODE1($LENGTH-1$) $                                    00024200        0258
 0220             DIRECT                                                      00024300        0259
 0221    L6     LDQ     HASWD                                                 00024400        0260
 0222           DIV     CW1LEN                                                00024500        0261
 0223           TPL     *+2                                                   00024600        0262
 0224           NEG                                                           00024700        0263
 0225           LDQ     HASWD                                                 00024800        0264
 0226           STQ     COMWD1+1,AL                                           00024900        0265
 0227             JOVIAL                                                      00025000        0266
 0228             TEST N $                                                    00025100        0267
 0229             DIRECT                                                      00025200        0268
 0230    L12    LDQ     HASWD                                                 00025300        0269
 0231           ADLQ    HASWD+1                                               00025400        0270
 0232           DIV     CW2LEN                                                00025500        0271
 0233           TPL     *+2                                                   00025600        0272
 0234           NEG                                                           00025610        0273
 0235           ALS     1                                                     00025800        0274
 0236           ERX0    COMWD2+1,AL                                           00026000        0275
 0237           LDAQ    HASWD                                                 00026100        0276
 0238           STX0    *+1                                                   00026200        0277
 0239           STAQ    **                                                    00026300        0278
 0240             JOVIAL                                                      00026400        0279
 0241            END ''N''                                                    00026500        0280
 0242            GOTO LOOP2 $                                                 00026600        0281
 0243    OUT1.   IF LENGTH EQ 1 $                                             00027300        0282
 0244            GOTO LOOP1 $                                                 00027400        0283
 0245            RETURN $                                                     00027500        0284
 0246           END ''PLNTAB''                                                00027600        0285
 0246    ''*****************************************************************''00028100        0286
 0246    ''                               SUPTAB                            ''00028200        0287
 0246    ''*****************************************************************''00028300        0288
 0247            CLOSE SUPTAB $                                               00028400        0289
 0248            BEGIN ''SUPTAB''                                             00028500        0290
 0249            LENGTH=0 $                                                   00028600        0291
 0250            HASWD=12H(000000000000) $                                    00028700        0292
 0251    LOOP5.  LENGTH=LENGTH+1 $                                            00028800        0293
 0252    LOOP6.  GETCRD(=CARD) $                                              00028900        0294
 0253            FOR N=0,1,11 $                                               00029000        0295
 0254             BEGIN ''N''                                                 00029100        0296
 0255              IF XCARD($N$) EQ O(545454202020) $                         00029200        0297
 0256              GOTO OUT2 $                                                00029300        0298
 0257              BYTE($0,6$)(HASWD)=XCARD($N$) $                            00029400        0299
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0258                     FOR I=0,1,5 $                                  00029500     0300
 0259                      BEGIN ''I''                                   00029600     0301
 0260                       IF BYTE($I,1$)(HASWD) NC 1H( ) $             00029700     0302
 0261                       TEST $                                       00029800     0303
 0262                       BYTE($I,1$)(HASWD)=1H(0) $                   00029900     0304
 0263                      END ''I''                                     00030000     0305
 0264                     GOTO MODE2($LENGTH-1$) $                       00030100     0306
 0265                     DIRECT                                         00030200     0307
 0266       L1      LDQ       HASWD                                      00030300     0308
 0267               DIV       ST1LEN                                     00030400     0309
 0268               TPL       *+2                                        00030500     0310
 0269               NEG                                                  00030600     0311
 0270               LDQ       HASWD                                      00030700     0312
 0271               STQ       SUP1TB+1,AL                                00030800     0313
 0272                     JOVIAL                                         00030900     0314
 0273                         TEST N $                                   00031000     0315
 0274                     DIRECT                                         00031100     0316
 0275       L2      LDQ       HASWD                                      00031200     0317
 0276               DIV       ST2LEN                                     00031300     0318
 0277               TPL       *+2                                        00031400     0319
 0278               NEG                                                  00031500     0320
 0279               LDQ       HASWD                                      00031600     0321
 0280               STQ       SUP2TB+1,AL                                00031700     0322
 0281                     JOVIAL                                         00031800     0323
 0282                         TEST N $                                   00031900     0324
 0283                     DIRECT                                         00032000     0325
 0284       L3      LDQ       HASWD                                      00032100     0326
 0285               DIV       ST3LEN                                     00032200     0327
 0286               TPL       *+2                                        00032300     0328
 0287               NEG                                                  00032400     0329
 0288               LDQ       HASWD                                      00032500     0330
 0289               STQ       SUP3TB+1,AL                                00032600     0331
 0290                     JOVIAL                                         00032700     0332
 0291                         TEST N $                                   00032800     0333
 0292                     DIRECT                                         00032900     0334
 0293       L4      LDQ       HASWD                                      00033000     0335
 0294               DIV       ST4LEN                                     00033100     0336
 0295               TPL       *+2                                        00033200     0337
 0296               NEG                                                  00033300     0338
 0297               LDQ       HASWD                                      00033400     0339
 0298               STQ       SUP4TB+1,AL                                00033500     0340
 0299                     JOVIAL                                         00033600     0341
 0300                    END ''N''                                       00033700     0342
 0301                   GOTO LOOP6 $                                     00033800     0343
 0302       OUT2.       IF LENGTH NQ 4 $                                 00036500     0344
 0303                   GOTO LOOP5 $                                     00036600     0345
 0304                    RETURN $                                        00036700     0346
 0305                  END '' SUPTAB''                                   00036800     0347
 0305       ''***************************************************''     00036900     0348
 0305       ''                    WORD                            ''     00037000     0349
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0305       ''*******************************************************************''   00037100             0350
  0306              CLOSE WORD $                                                      00037200             0351
  0307               BEGIN ''WORD''                                                   00037300             0352
  0308                COUNT=0 $                                                       00037400             0353
  0309       WORD0. IO022(0,INTAPE,0,ABSTOR,500) $                                    00037500             0354
  0310              WAIT(IO022) $                                                     00037600             0355
  0311              DIRECT                                                            00037610             0356
  0312              AOS     DOCNUM                                                    00037611             0357
  0313              JOVIAL                                                            00037612             0358
  0314              IF INTAPE EQ V(EOF) $                                             00037700             0359
  0315               BEGIN                                                            00037800             0360
  0316                IO080(3,PRNT,0,DOCNUM,1) $                                      00037820             0361
  0317                RETURN $                                                        00037830             0362
  0318               END                                                              00038000             0363
  0318       ''     NOW RESET TALLY COUNTERS FOR BYTE FETCHES ''                      00038200             0364
  0319              DIRECT                                                            00038300             0365
  0320              LDA     TALLY3          RESET                                     00038400             0366
  0321              STA     TALLY4          TALLY COUNT                               00038500             0367
  0322              LDA     TALLY5                                                    00038600             0368
  0323              STA     TALLY6          RESET THIS ONE TOC                        00038700             0369
  0324       ENDWRD LDA     TALLY4,SC       FETCH A BYTE                              00038800             0370
  0325              TTF     PEEKTB+1,AL                                               00038900             0371
  0326              TRA     PROCES          CHARACTER OVERFLOW                        00038910             0372
  0327       MOVEBY STA     TALLY6,SC       STORE A BYTE                              00039000             0373
  0328              TTF     ENDWRD          STORE                                     00039100             0374
  0329              TRA     WORD2           TREAT WORD AS COMMON                      00039110             0375
  0330       WORD1  LDA     TALLY6                                                    00039400             0376
  0331              ALS     18                                                        00039500             0377
  0332              ARL     24                                                        00039600             0378
  0333              STA     COUNT                                                     00039700             0379
  0334              LDQ     60,DL                                                     00039710             0380
  0335              SBQ     COUNT                                                     00039720             0381
  0336              SZN     COUNT                                                     00039730             0382
  0337              TZE     ENDWRD                                                    00039740             0383
  0338              JOVIAL                                                            00039799             0384
  0339               IFEITH COUNT GR 18 $                                             00039900             0385
  0340               GOTO WORD2 $                                                     00039910             0386
  0341               ORIF COUNT GR 12 $                                               00039920             0387
  0342               GOTO LOOP12 $ ''CAN'T BE COMMON/BEGIN SUFFIX REMOVAL''           00040000             0388
  0343               ORIF COUNT GR 6 $                                                00040200             0389
  0344               GOTO PLEBE2 $                                                    00040205             0390
  0345               ORIF COUNT EQ 1 $                                                00040210             0391
  0346               GOTO WORD2 $                                                     00040220             0392
  0347               END                                                              00040300             0393
  0348              DIRECT                                                            00040400             0394
  0349              LDQ     WUNWRD                                                    00040500             0395
  0350              DIV     CW1LEN                                                    00040600             0396
  0351              TPL     *+2             BRANCH IF REMAINDER NEGATIVE              00040700             0397
  0352              NEG                                                               00040800             0398
  0353              LDQ     COMWD1+1,AL     LOAD WITH HASH-FOUND ADDRESS              00041100             0399
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0354          CMPQ   WUNWRD        COMPARE WITH THE WORD HASHED                 00041200      0400
 0355          TZE    WORD2         IF FOUND, IT'S COMMON                        00041210      0401
 0356          LDA    COUNT         GET CHAR COUNT                               00041220      0402
 0357          CMPA   4,DL          SEE IF IT CAN BE STEMMED                     00041230      0403
 0358          TZE    ROOT8         IT CAN'T                                     00041240      0404
 0359          TMI    ROOT8         IT CAN'T                                     00041250      0405
 0360          TRA    LOOP12        IT CAN                                       00041300      0406
 0361   PLEBE2 LDQ    WUNWRD                                                     00041400      0407
 0362          ADLQ   WUNWRD+1                                                   00041500      0408
 0363          DIV    CW2LEN                                                     00041600      0409
 0364          TPL    *+2           BRANCH IF REMAINDER NEGATIVE                 00041700      0410
 0365          NEG                                                               00041800      0411
 0366          ALS    1                                                          00041900      0412
 0367          EAX7   COMWD2+1,AL   CALC & SAVE HASH-FOUND ADDRESS               00042100      0413
 0368          LDAQ   0,7           LOAD WITH HASH-FOUND ADDRESS                 00042200      0414
 0369          CMPAQ  WUNWRD        COMPARE WITH WORD HASHED                     00042300      0415
 0370          TZE    WORD2                                                      00042400      0416
 0371   LOOP12 LDA    5,DL                                                       00042600      0417
 0372          STA    SUB           PRESET SUB TO FIVE                           00042700      0418
 0373   LOOP13 LDA    SUB           LOAD SUFFIX LENGTH TALLY COUNTER             00042799      0419
 0374          SBA    1,DL          DECREMENT BY ONE                             00042800      0420
 0375          STA    SUB           RESAVE DECREMENTED COUNTER                   00042810      0421
 0376          LDA    0,DL                                                       00042900      0422
 0377          STA    SUFFIX        INITIALIZE THE SUFFIX WORD                   00042910      0423
 0378          JOVIAL                                                            00042920      0424
 0379               BYTE($0,SUB$)(SUFFIX)=BYTE($COUNT-SUB,SUB$)(WUNWRD) $        00043000      0425
 0380          DIRECT                                                            00043200      0426
 0381          LDA    SUB           LOAD SUFFIX INDEX COUNTER                    00043210      0427
 0382          SBA    1,DL          DECREMENT                                    00043220      0428
 0383          TSX1   MODE10        TRANSFER ACCORDING TO SUFFIX LENGTH          00043240      0429
 0384   ROOT1  LDQ    SUFFIX                                                     00043400      0430
 0385          DIV    ST1LEN                                                     00043500      0431
 0386          TPL    *+2           BRANCH IF REMAINDER NEGATIVE                 00043600      0432
 0387          NEG                                                               00043700      0433
 0388          LDQ    SUF1TB+1,AL   LOAD WITH CONTENTS OF HASHED ADDRESS         00043800      0434
 0389          CMPQ   SUFFIX        COMPARE WITH SUFFIX                          00043900      0435
 0390          TZE    ROOT6                                                      00044110      0436
 0391          TRA    ROOT8         SUFFIX REMOVAL COMPLETED                     00044200      0437
 0392   ROOT2  LDQ    SUFFIX                                                     00044300      0438
 0393          DIV    ST2LEN                                                     00044400      0439
 0394          TPL    *+2                                                        00044500      0440
 0395          NEG                                                               00044600      0441
 0396          LDQ    SUF2TB+1,AL   LOAD WITH CONTENTS OF HASHED ADDRESS         00044700      0442
 0397          CMPQ   SUFFIX        COMPARE WITH SUFFIX                          00044800      0443
 0398          TZE    ROOT6                                                      00045010      0444
 0399          TRA    LOOP13                                                     00045100      0445
 0400   ROOT3  LDQ    SUFFIX                                                     00045200      0446
 0401          DIV    ST3LEN                                                     00045300      0447
 0402          TPL    *+2           BRANCH IF REMAINDER NEGATIVE                 00045400      0448
 0403          NEG                                                               00045500      0449
```

Figure V.15a. (Continued)

```
1321T 01   10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

    0404             LDQ     SUP3TB+1,AL      LOAD WITH CONTENTS OF HASHED ADDRESS    00045600          0450
    0405             CMPQ    SUFFIX           COMPARE WITH SUFFIX                     00045800          0451
    0406             TZE     ROOT6                                                    00046010          0452
    0407             TRA     LOOP13                                                   00046100          0453
    0408    ROOT4    LDQ     SUFFIX                                                   00046200          0454
    0409             DIV     ST4LEN                                                   00046300          0455
    0410             TPL     *+2              BRANCH IF REMAINDER NEGATIVE            00046400          0456
    0411             NEG                                                              00046500          0457
    0412             LDQ     SUP4TB+1,AL      LOAD WITH CONTENTS OF HASHED ADDRESS    00046600          0458
    0413             CMPQ    SUFFIX           COMPARE WITH SUFFIX                     00046700          0459
    0414             TNZ     LOOP13                                                   00047000          0460
    0415             JOVIAL                                                           00047100          0461
    0416    ROOT6.         IF COUNT-SUB GQ 4 $                                        00047200          0462
    0417                     BEGIN                                                    00047300          0463
    0418                       SUFFIX=6H(000000) $                                    00047400          0464
    0419                       COUNT=COUNT-SUB $                                      00047500          0465
    0420                       GOTO LOOP12 $                                          00047600          0466
    0421                     END                                                      00047700          0467
    0422                 IF COUNT GQ 5 $                                              00047702          0468
    0423                 GOTO LOOP13 $                                                00047705          0469
    0424    ROOT8.       IF COUNT GR 12 $                                             00049150          0470
    0425                 GOTO WORD2 $                                                 00049160          0471
    0426                 IF BYTE($COUNT-2,1$)(WUNWRD) EQ BYTE($COUNT-1,1$)(WUNWRD) $  00049165          0472
    0427                   BEGIN                                                      00049175          0473
    0428                     NXBYTE=BYTE($COUNT-1,1$)(WUNWRD) $                       00049180          0474
    0429                     DIRECT                                                   00049185          0475
    0430             LDA       NXBYTE         GET THE BYTE                            00049190          0476
    0431             SZN       BLIM+1,AL      LOAD INDICATOR WITH CHARACTER IN BLIM   00049195          0477
    0432             TZE       BTS0           TRANSFER IF IT IS A VOWEL               00049200          0478
    0433             LDA       COUNT          GET CHARACTER COUNT                     00049205          0479
    0434             SBA       1,DL           DECREMENT                               00049210          0480
    0435             STA       COUNT          STORE COUNT                             00049215          0481
    0436                     JOVIAL                                                   00049220          0482
    0437                   END                                                        00049223          0483
    0438    BTS0.        BYTE($0,COUNT$)(STEM)=BYTE($0,COUNT$)(WUNWRD) $              00049225          0484
    0439             DIRECT                                                           00049400          0485
    0440             LDQ     STEM+1           LOAD 2ND WORD OF STEM                   00049500          0486
    0441             TNZ     BTS3             NOT ZERO/LOGICAL ADD NECESSARY          00049600          0487
    0442             LDQ     STEM             USE ONLY 1ST HALF OF STEM               00049610          0488
    0443    BTS4     DIV     RTLEN            DIVIDE BY LENGTH OF ROOTS TABLE         00049700          0489
    0444             TPL     *+2                                                      00049800          0490
    0445             NEG                                                              00049900          0491
    0446             AOS     WGHT+1,AL                                                00049904          0492
    0447             ALS     1                                                        00050000          0493
    0448             EAX7    ROOTS+1,AL                                               00050200          0494
    0449             SZN     0,7                                                      00050300          0495
    0450             TNZ     BTS2                                                     00050800          0496
    0451             AOS     UNQRTS                                                   00050900          0497
    0452             LDAQ    STEM                                                     00050901          0498
    0453             STAQ    0,7                                                      00051000          0499
```

Figure V. 15a. (Continued)

```
1321T 01  10-09-70   13.463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0454     RTS2   AOS    RTCNT                                                    00051200        0500
  0455            TRA    WORD2                                                    00051210        0501
  0456     RTS3   ADLQ   STEM          DO LOGICAL ADD  SINCE COUNT>6              00051220        0502
  0457            TRA    RTS4          PROCEED TO DO THE DIVISION                 00051230        0503
  0458     WORD2  STZ    COUNT                                                    00051300        0504
  0459            LDAQ   EMPTY         STORE ZERO IN AQ                           00051400        0505
  0460            STAQ   STEM                                                     00051430        0506
  0461            STZ    WUNWRD        CLEAR TEMP STORAGE                         00051600        0507
  0462            STZ    WUNWRD+1      FOR                                        00051700        0508
  0463            STZ    WUNWRD+2      ISOLATED WORD                              00051800        0509
  0464            STZ    WUNWRD+3                                                 00051802        0510
  0465            STZ    WUNWRD+4                                                 00051804        0511
  0466            STZ    WUNWRD+5                                                 00051806        0512
  0467            STZ    WUNWRD+6                                                 00051807        0513
  0468            STZ    WUNWRD+7                                                 00051808        0514
  0469            STZ    WUNWRD+8                                                 00051809        0515
  0470            STZ    WUNWRD+9                                                 00051810        0516
  0471            LDA    TALLY5        RESET TALLY                                00051900        0517
  0472            STA    TALLY6        COUNTER                                    00052000        0518
  0473            TRA    ENDWRD                                                   00052100        0519
  0474     PROCES STZ    SLOT                                                     00052200        0520
  0475            STZ    LINT                                                     00052210        0521
  0476     PACKO  LDA    LINT                                                     00052220        0522
  0477            EAX7   WGHT+1,AL     CAL ADDRESS IN WEIGHTS TABLE               00052225        0523
  0478            ALS    1             MULTIPLY BY TWO FOR CORRECT ADDRESS        00052230        0524
  0479            EAX6   ROOTS+1,AL    CALCULATE ADDRESS IN ROOTS TABLE           00052240        0525
  0480            LDAQ   0,6           GET THE ROOT                               00052250        0526
  0481            TZE    PACKS         IF EMPTY, GO TO NEXT ADDRESS               00052270        0527
  0482            LDA    SLOT                                                     00052290        0528
  0483            EAX5   WGHT+1,AL     CALC PACKED WEIGHT LOCATION                00052292        0529
  0484            ALS    1             MULT BY TWO FOR CORRECT REL. ADDRESS       00052300        0530
  0485            EAX4   ROOTS+1,AL    CALC PACKED ROOT LOCATION                  00052310        0531
  0486            LDAQ   0,6           GET UNPACKED ROOT                          00052320        0532
  0487            STAQ   0,4           STORE ROOT IN PACKED LOCATION              00052330        0533
  0488            STZ    0,6           ZERO IN THE                                00052332        0534
  0489            STZ    1,6           UNPACKED ROOT                              00052333        0535
  0490            LDA    0,7           GET THE UNPACKED WEIGHT                    00052340        0536
  0491            STA    0,5           STORE WEIGHT IN PACKED LOCATION            00052350        0537
  0492            STZ    0,7           ZERO UNPACKED WEIGHT                       00052352        0538
  0493            AOS    SLOT          INCREMENT PACKING FACTOR                   00052450        0539
  0494            LDA    UNQRTS        GET COUNTER OF UNIQUE ROOTS                00052460        0540
  0495            CMPA   SLOT          SEE IF ALL ROOTS ARE PACKED                00052470        0541
  0496            TZE    SRTO          IF SO, GO TO SORTING ROUTINE               00052480        0542
  0497     PACKS  AOS    LINT                                                     00052490        0543
  0498            LDA    2532,DL                                                  00052500        0544
  0499            CMPA   LINT                                                     00052510        0545
  0500            TPL    PACKO                                                    00052530        0546
  0501               JOVIAL                                                       00052540        0547
  0502     SRTO.       LL=2 $                                                     00053600        0548
  0503     SRT1.         IF LL GO UNQRTS $                                        00053700        0549
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13.463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0504                 GOTO SRT2 $                                                       00053800          0550
 0505                 LL=LL*2      $                                                    00053900          0551
 0506                 GOTO SRT1 $                                                       00054000          0552
 0507       SRT2.     DD=(LL/2)-1 $                                                     00054100          0553
 0510       SRT3.     II=0 $ JJ=0 $ KK=0 $                                              00054200          0554
 0511                 DIRECT                                                            00054300          0555
 0512       SRT4   LDA      II                                                          00054400          0556
 0513              EAX7     WGHT+1,AL                                                   00054410          0557
 0514              ALS      1                                                           00054500          0558
 0515              EAX6     ROOTS+1,AL                                                  00054710          0559
 0516              LDAQ     0,6                                                         00054900          0560
 0517              STAQ     W2                                                          00055000          0561
 0518              LDA      II                                                          00055100          0562
 0519              ADA      DD                                                          00055200          0563
 0520              EAX5     WGHT+1,AL                                                   00055210          0564
 0521              ALS      1                                                           00055300          0565
 0522              EAX4     ROOTS+1,AL                                                  00055610          0566
 0523              LDAQ     0,4                                                         00055700          0567
 0524              CMPAQ    W2                                                          00055800          0568
 0525              TNC      SRT7                                                        00055900          0569
 0526                 JOVIAL                                                            00056000          0570
 0527       SRT5,  IF II+DD+1 LQ UNQRTS $                                               00056100          0571
 0528                 BEGIN                                                             00056200          0572
 0529                 II=II+1 $                                                         00056300          0573
 0530                 GOTO SRT4 $                                                       00056400          0574
 0531                 END                                                               00056500          0575
 0532                 DD=(DD-1)/2 $                                                     00056600          0576
 0533                 IF DD LQ 0 $                                                      00056700          0577
 0534                 GOTO SRT13 $                                                      00056800          0578
 0535                 GOTO SRT3 $                                                       00056810          0579
 0536               DIRECT                                                              00056900          0580
 0537       SRT7   LDAQ     0,6                                                         00056910          0581
 0538              STAQ     W2                 STORE IN W2                              00056920          0582
 0539              LDAQ     0,4                                                         00056930          0583
 0540              STAQ     0,6                                                         00056940          0584
 0541              LDAQ     W2                 LOAD 1ST ROOT                            00056950          0585
 0542              STAQ     0,4                                                         00056960          0586
 0543              LDA      0,7                                                         00056970          0587
 0544              STA      TEMP               STORE IN TEMP                            00056980          0588
 0545              LDA      0,5                                                         00056990          0589
 0546              STA      0,7                                                         00057000          0590
 0547              LDA      TEMP               LOAD 1ST WEIGHT                          00057010          0591
 0548              STA      0,5                                                         00057020          0592
 0549               JOVIAL                                                              00057030          0593
 0550                 JJ=II-DD $                                                        00057100          0594
 0551                 IF JJ LS 0 $                                                      00057200          0595
 0552                 GOTO SRT5 $                                                       00057300          0596
 0553                 KK=II $                                                           00057400          0597
 0554                 DIRECT                                                            00057500          0598
 0555       SRT8   LDA      JJ                                                          00057600          0599
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13.463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

0556            EAX7    WGHT+1,AL                                   00057610         0600
0557            ALS     1                                           00057700         0601
0558            EAX6    ROOTS+1,AL                                  00058010         0602
0559            LDAQ    0,6                                         00058100         0603
0560            STAQ    W2                                          00058200         0604
0561            LDA     KK                                          00058300         0605
0562            EAX5    WGHT+1,AL                                   00058310         0606
0563            ALS     1                                           00058400         0607
0564            EAX4    ROOTS+1,AL                                  00058710         0608
0565            LDAQ    0,4                                         00058800         0609
0566            CMPAQ   W2                                          00058900         0610
0567            TZE     SRT5                                        00059000         0611
0568            TRC     SRT5                                        00059010         0612
0569            LDAQ    0,6                                         00059100         0613
0570            STAQ    W2                                          00059110         0614
0571            LDAQ    0,4                                         00059120         0615
0572            STAQ    0,6                                         00059130         0616
0573            LDAQ    W2                                          00059140         0617
0574            STAQ    0,4                                         00059150         0618
0575            LDA     0,7                                         00059160         0619
0576            STA     TEMP                                        00059170         0620
0577            LDA     0,5                                         00059180         0621
0578            STA     0,7                                         00059190         0622
0579            LDA     TEMP                                        00059200         0623
0580            STA     0,5                                         00059210         0624
0581              JOVIAL                                            00059300         0625
0582                JJ=JJ-DD $                                      00059400         0626
0583                IF JJ LS 0 $                                    00059500         0627
0584                GOTO SRT5 $                                     00059600         0628
0585                KK=JJ+DD $                                      00059700         0629
0586                GOTO SRT8 $                                     00059800         [illegible]630
0587        SRT13, SLOT=0 $                                         00059910         0631
0588            FOR N=0,1,UNQRTS-1 $                                00059920         0632
0589             BEGIN ''N''                                        00059925         0633
0590              DIRECT                                            00059926         0634
0591            LDA     N                                           00059928         0635
0592            EAX7    WGHT+2,AL       CALC ADDRESS OF WEIGHT      00059930         0636
0593            ALS     1               MULT REL ADDRESS BY TWO     00059940         0637
0594            EAX6    ROOTS+3,AL      CALC ADDRESS OF ROOTS       00059950         0638
0595            LDAQ    0,6             LOAD THE ROOT               00059970         0639
0596            STAQ    CONCRD+1        STORE IN OUTPUT TABLE       00059980         0640
0597            STZ     0,6             ZERO IN                     00059985         0641
0598            STZ     1,6             THE ROOT                    00059990         0642
0599            LDA     DOCNUM          GET THE DOCUMENT NUMBER     00060000         0643
0600            STA     CONCRD+3        STORE IN CONCORDANCE OUTPUT TABLE 00060010   0644
0601            LDA     0,7             GET THE WEIGHT              00060020         0645
0602            STA     CONCRD+4        STORE IN CONCORDANCE OUTPUT TABLE 00060040   0646
0603            STZ     0,7             ZERO THE WEIGHT             00060050         0647
0604              JOVIAL                                            00060060         0648
0605            IO050(4,CONC,0,CONCRD,4) $                          00060070         0649
```

Figure V.15a. (Continued)

```
1321T 01  10-09-70   13,463   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0606              END ''N''                                                    00060120        0650
 0607             IO055(4,LEADER,0,BTCNT,1) $                                   00060130        0651
 0608               COUNT=0 $                                                   00060400        0652
 0609               BTCNT=0 $                                                   00060405        0653
 0610               UNQRTS=0 $                                                  00060410        0654
 0611               GOTO WORDO $                                                00060430        0655
 0612              END ''WORD''                                                 00060500        0656
 0612      ''*****************************************************************''  00062700        0657
 0612      ''                        GETCRD                                     ''  00062800        0658
 0612      ''*****************************************************************''  00062900        0659
 0612      ''*****************************************************************''  00063000        0660
 0613             PROC GETCRD(=CARD) $                                          00063100        0661
 0614             TABLE CARD R 14 S 1 $                                         00063200        0662
 0614              BEGIN '' CARD''                                              00063300        0663
 0615               ITEM XCARD H 6 0 0 $                                        00063400        0664
 0616              END ''CARD''                                                 00063500        0665
 0616              BEGIN ''GETCRD''                                             00063800        0666
 0618               IO001(0,REDER,0,CARD,14) $ WAIT(IO001) $                    00063900        0667
 0619              END ''GETCRD''                                               00068600        0668
 0620             TERM GO $
```

Figure V.15a. (Concluded)

V.5 THE STATISTICAL FILTER

The basic tenet of this retrieval system is that a document collection can be characterized by a limited number of tokens, designated for the purposes of this report as the content stems. The set of content stems is representative of the different semantic areas present in the document collection. However, it is not a matter of trivial concern to separate content from non-content tokens over the collection. If each document in the collection is to be characterized by the weighted presence of these content stems, some reliable means must be devised to perform this separation. Such a method has been devised by Sally Dennis.(2)

V.5.1 Derivation of the Statistical Filter Algorithm

Dennis determines a single function with a value determined by the number of occurrences of the word in the entire collection multiplied by the normalized variation of within-document frequency. This is generally referred to (for the c^{th} word) as $NOCC/EK_c$.* The details of the computation $NOCC/EK_c$ follow. Dennis' measure is called V_c, as it is convenient to use a different form from that of Dennis. The computed values are, however, the same as hers. The primary concern here is to compute the measure for a large number of stems, while reducing the required number of calls on auxiliary storage devices.

Let D be the number of documents in the collection, and L_d be the number of stems in document d (that is, the length of document d). The number of occurrences of stem c in document d is $f_{c,d}$, and its normalized form is $g_{c,d} = f_{c,d}/L_d$. The normalized frequency of occurrences over the collection is:

$$\overline{g}_c = \sum_{d=1}^{D} g_{c,d}/D.$$

The number of raw occurrences of stem c in the collection is σ_c, where

---

* The symbol NOC_c/EK_c, although a departure from the original terminology, is perhaps clearer.

$$\sigma_c = \sum_{d=1}^{D} f_{c,d}.$$

Letting $s_c^2 = \sum_{d=1}^{D} (g_{c,d} - \bar{g}_c)^2 /(D-1)$ then Dennis' measure is given by $\sigma_c s_c^2/\bar{g}_c^2$, and is equal to the present V_c. In order that V_c can be computed in one pass through the vector $[f_{c,1}\ f_{c,2} \ldots f_{c,D}]$, some algebraic manipulations are necessary.

Knowing the total number of documents in the collection and assuming that a table giving only the total number of raw stems in each document can be kept in high speed storage, we wish to compute V_c for every raw stem making only one pass through the concordance. It is assumed that the concordance vectors for any stem can be kept in core, but that only one such vector can be kept in core at any one time. If V_c exceeds some limit, the corresponding raw stem is said to be a content stem, and can be written on a tape of content stems along with its concordance vector. Thus, $L_1, L_2, \ldots, L_d, \ldots, L_D$ are available in core. A sequential tape ordered on c and containing $f_{c,1}, f_{c,2}, \ldots, f_{c,d}, \ldots, f_{c,D}$ for every c is available, but only the values of $f_{c,d}$ for one c and for $d=1, \ldots, D$ can be held in core at a given time.

Let the normalized number of occurrences of stem c in document d be

$$g_{c,d} = f_{c,d}/L_d, \tag{1}$$

and the mean value be

$$\bar{g}_c = \sum_{d=1}^{D} g_{c,d}/D. \tag{2}$$

The total number of occurrences of stem c in the collection is

$$\sigma_c = \sum_{d=1}^{D} f_{c,d}. \tag{3}$$

The variance of the $g_{c,d}$ is

$$s_c^2 = \sum_{d=1}^{D} (g_{c,d} - \bar{g}_c)^2/(D-1) \tag{4}$$

Yielding for Dennis' measure

$$V_c = \sigma_c s_c^2/\bar{g}_c^2 \tag{5}$$

In order to compute V_c with only one pass through the $f_{c,d}$, two additional variables are introduced, in terms of the data from which they are computed:

$$\alpha_c = \sum_{d=1}^{D} (f_{c,d}/L_d)^2 \ , \tag{6}$$

and

$$\beta_c = \sum_{d=1}^{D} f_{c,d}/L_d. \tag{7}$$

Note that

$$\alpha_c = \sum_{d=1}^{D} (g_{c,d})^2 , \tag{8}$$

and

$$\beta_c = \sum_{d=1}^{D} g_{c,d} \tag{9}$$

So from Eq. 2 and Eq. 9, it follows that

$$\bar{g}_c = \beta_c/D. \tag{10}$$

Now, upon substituting Eq. 4 into 5 in order to obtain an intermediate form for V_c, there obtains:

$$V_c = \frac{\sigma_c}{\bar{g}_c^2} \quad \frac{\sum_{d=1}^{D} (g_{c,d} - \bar{g}_c)^2}{D-1} \quad . \tag{11}$$

Expanding the squared term in the numerator and applying Eq. 8 and Eq. 10 gives,

$$V_c = \frac{\sigma_c}{(\beta_c/D)^2} \cdot \frac{\alpha_c - 2\frac{\beta^2}{D} \sum_{d=1}^{D} g_{c,d} + D(\frac{\beta^2}{D})^2}{D-1} \tag{12}$$

But β_c/D is independent of d and as $D >> 1$, $D^2/(D-1) \approx D$. Thus applying Eq. 9 one obtains,

$$V_c = \frac{\sigma_c D}{\beta_c^2} \left[\alpha_c - 2\frac{\beta_c}{D}(\beta_c) + D(\frac{\beta_c}{D})^2 \right] \tag{13}$$

or

$$V_c \approx \frac{\sigma_c D}{\beta_c^2} \left[\alpha_c - \frac{\beta_c^2}{D} \right] \tag{14}$$

or

$$V_c \approx \sigma_c \left[\alpha_c D/\beta_c^2 - 1 \right] \tag{15}$$

This can be computed in one pass through the concordance vector for each stem c, with the aid of a table of values of L_d.

It is clear from the above discussion that once all isolated stems for the collection have been passed through the statistical filter, each stem will be characterized by a V_c value. After the stems have been ranked by descending V_c value the task remainds to determine what the cut-off will be between content and non-content stems.

V. 5. 2 Establishing the Content-Stem Cut-off Levels

In this sub-section two topics are approached:

1) characteristics of Dennis' measure which aid in establishing the cut-off levels
2) experimental observations on actual data.

V. 5. 2. 1 A Property of Dennis' Measure. In Dennis' notation, consider the situation when a token occurs once in a document collection. If this token is identified by c* and it occurs in document d*, then:

$$f_{c^*,d} = 0 \text{ for } d \neq d^*,$$

$$f_{c^*,d^*} = 1,$$

$$g_{c^*,d} = 0 \text{ for } d \neq d^*, \text{ and}$$

$$g_{c^*,d^*} = 1/L_{d^*}.$$

By definition[2],

$$\bar{g}_c = \sum_{d=1}^{N} g_{c,d}/N, \text{ so}$$

$$\bar{g}_{c^*} = 1/L_{d^*}N.$$

Again from the definitions in the original paper,

$$s_c^2 = \sum_{d=1}^{N} (g_{c,d} - \bar{g}_c)^2/(N-1).$$

Expanding the squared term and substituting for the case $c = c^*$,

$$s_{c^*}^2 = [(1/L_{d^*})^2 - 2(1/L_{d^*})(1/L_{d^*}N) + N(1/L_{d^*}N)^2]/[N-1],$$

or

$$s_{c^*}^2 = L_{d^*}^{-2}(1-N^{-1})/(N-1), \text{ which reduces to}$$

$$s_{c^*}^2 = 1/L_{d^*}^2 N.$$

Now, EK_c is defined as g_c^2/s_c^2, so for $c=c^*$ we have

$$EK_{c^*} = [1/L_{d^*}N]^2/[1/L_{d^*}^2 N], \text{ or finally}$$

$$EK_{c^*} = 1/N.$$

In the terms of the operational statistical filter package, it is apparent that stems for which σ_c is very low must be disqualified, since the corresponding V_c can be elevated by the small value of EK_c. It is interesting to note that the illustrations in Dennis' paper do not show EK_c smaller than about 0.002. Upon a little reflection, this is not surprising: EK_c reflects on the distribution of word c, and it only occurs once. Thus, (in Dennis' paper NOCC is the total number of raw occurrences; σ_c has been used in this report).

$V_c = NOCC_c/EK_c$, and $NOCC_{c*} = 1$. Therefore,

$V_{c*} = N$.

How this property of Dennis' measure affects the cut-off point is discussed in the next subsection.

V.5.2.2 Establishing Cut-off: Experimental Observations. It was noted in the discussion above that as σ_c approaches unity, $\bar{g}_c^2/s_c^2$ approaches $1/N$, so that $V_c = \sigma_c s_c^2/\bar{g}^2$ approaches N. It was then suggested that, in addition to all the area to the right of some line V_c = constant in the σ_c vs $\bar{g}_c^2/s_c^2$ plot, an additional area below some line of constant σ_c be excluded. Thus, only stems falling in the unshaded area of Figure V.16 would be passed by the statistical filter.

Figure V.17 appears to support this; for about half of the stems (the "tail" of the distribution) V_c is about equal to N, and the number of occurrences in the entire collection is one or two.

The number of raw stems encountered is expected to rise rapidly at first, and then more slowly, as additional documents are introduced. For any collection, many special tokens are expected to be encountered only once in the collection; many others are expected to be relatively infrequently encountered. In terms of V_c, this means many words will appear at the low end of the distribution when a small document collection is considered, but will move up as more documents are processed. Note that a few stems have quite high V_c values, and that this is followed by a sudden drop.

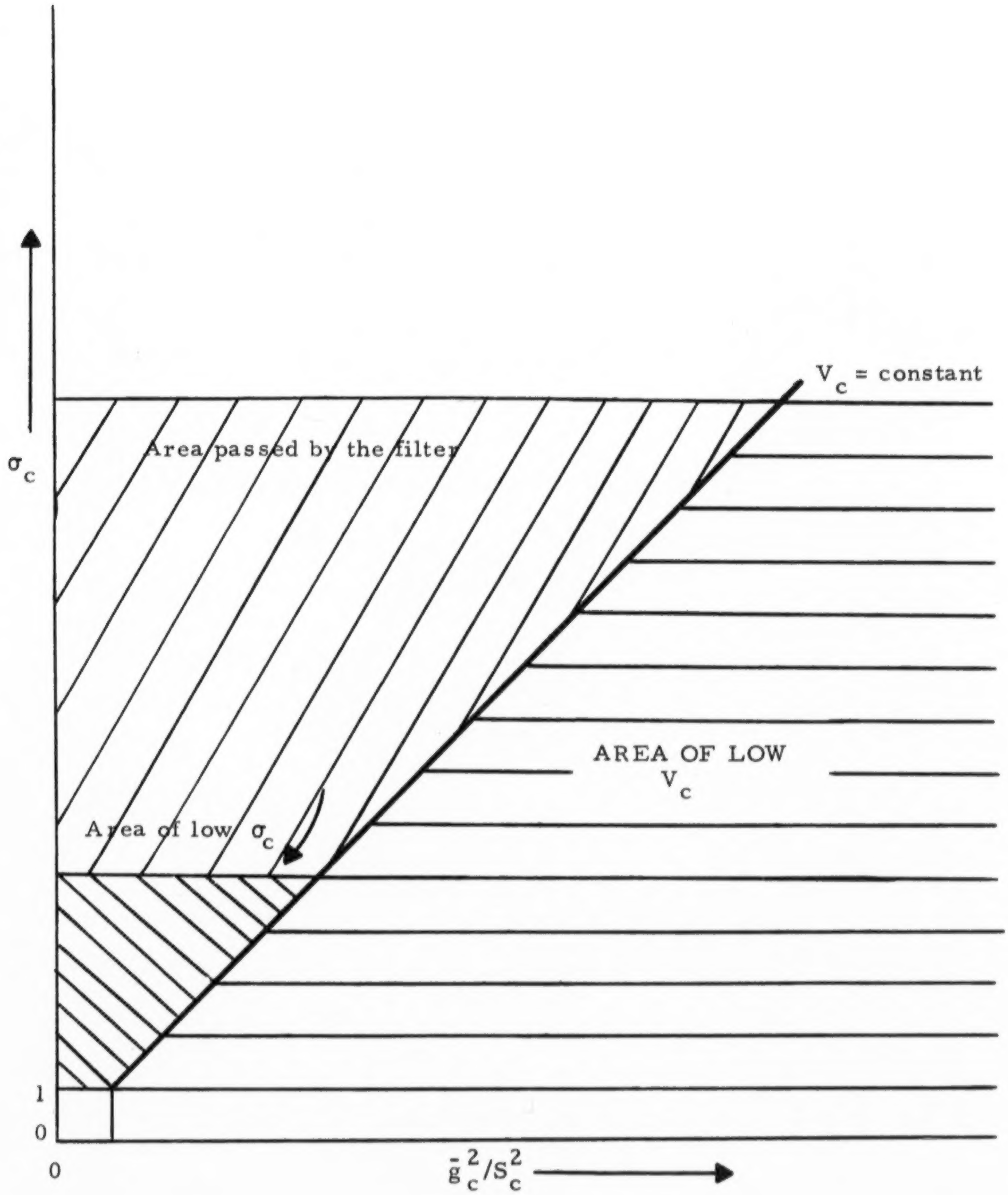

V_c = constant
Area passed by the filter
σ_c
AREA OF LOW V_c
Area of low σ_c
1
0
0
$\bar{g}_c^2/S_c^2$

Figure V. 16. Suggested V_c vs. $\bar{g}_c^2/S_c$ Cutoff Areas

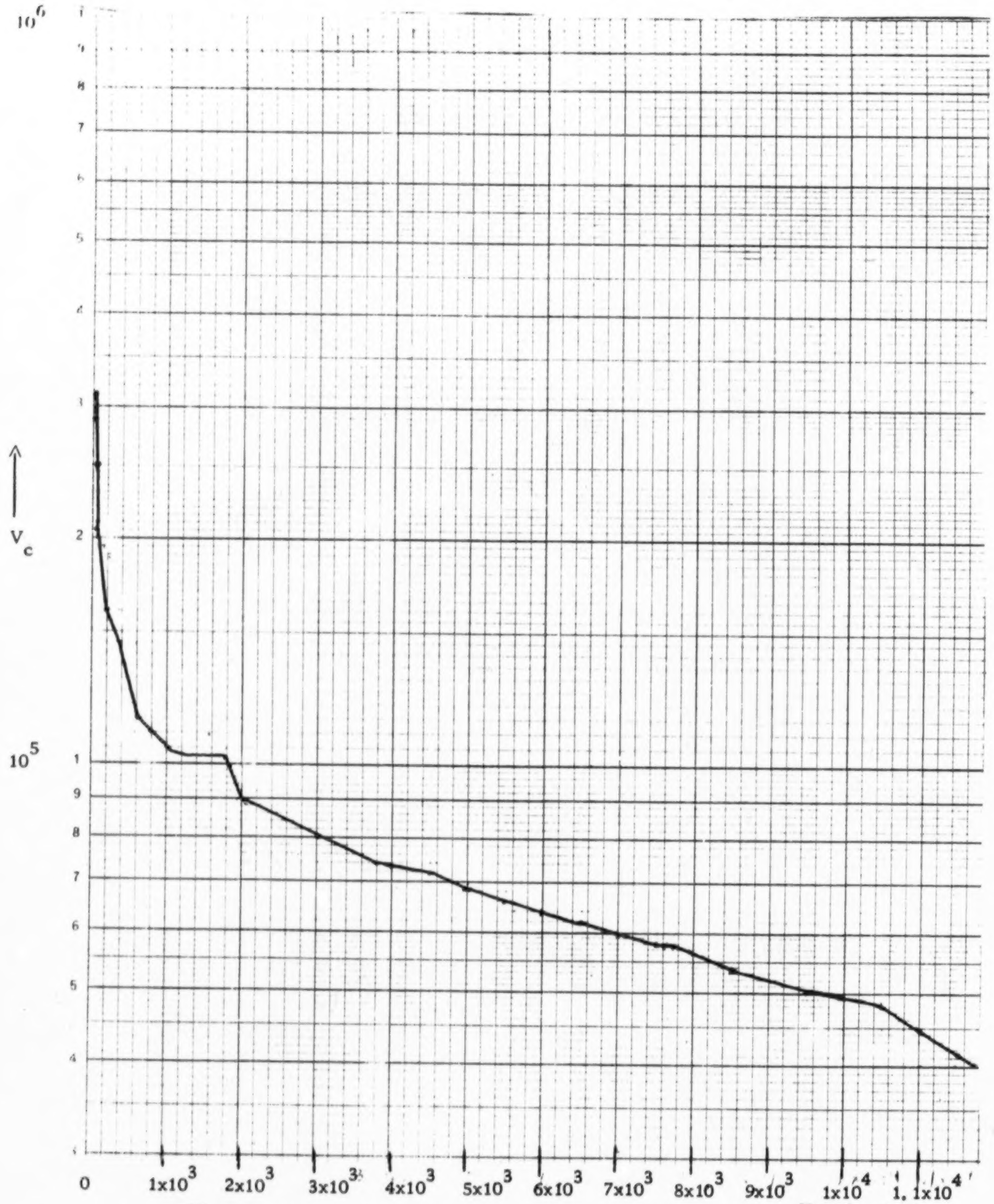


Figure V. 17. Stem Number By Rank of V_c

Jargon of the Collection

In observing the distributions obtained for the abstracts, several terms of questionable surface validity as content stems were found to have quite high values of V_c. Inspection of a few abstracts reveals that many special abbreviations are used in the data. Such abbreviations are, of course, useful for retrieval in many cases. But they certainly do not make the output from the statistical filter obvious.

An example is quoted below:

"IN EXP. I, VERBAL DISCRIMINATION (VD) LISTS COMPOSED OF HOMONYM WORD PAIRS AS WRONG (W) AND RIGHT (R) ITEMS WERE MORE DIFFICULT TO LEARN THAN A CONTROL LIST CONTAINING NO HOMONYM WORD PAIRS, SS WERE 36 UNDERGRADUATES, MOREOVER, AN INTRAPAIR W-R HOMONYM CONDITION WAS LESS DIFFICULT TO LEARN THAN AN INTERPAIR W-R CONDITION, FREE RECALL MEASURES OF W AND R ITEMS INDICATED THAT THE INTRA- AND INTERPAIR CONDITIONS DIFFERED PRIMARILY IN THE PROCESS EMPLOYED TO "TAG AND COLLAPSE" PAIRS OF ITEMS FOR USE IN THE RECOGNITION OF R ITEMS DURING VD PRACTICE, IN EXP. II, THE W 1-R 1 , W 2-R 1' VD TRANSFER PARADIGM WAS EMPLOYED WITH HOMONYM WORDS PROVIDING THE R 1-R 1' RELATIONSHIP. POSITIVE TRANSFER, RELATIVE TO THE W 1-R 1, W 2-R 2 CONTROL, WAS FOUND, BUT ONLY WHEN SS WERE INFORMED THAT LIST 2 CONTAINED ITEMS SIMILAR TO LIST 1 ITEMS."

Result Using Dennis' Measure

Figure V.18 and Table V.1 show selected points for the final run. Note that a line of constant ratio of σ_c to $\bar{g}_c^2/s_c^2$ is any line passing through the plot from lower left to upper right, at 45° angle. Since Dennis worked with greatly truncated word tokens (rather than stems), and with documents many times longer than the abstracts considered here, there is no reason to expect any special significance to be attached to her figure of 10,500 for a value of V_c (or NOCC/EK, in her terms).

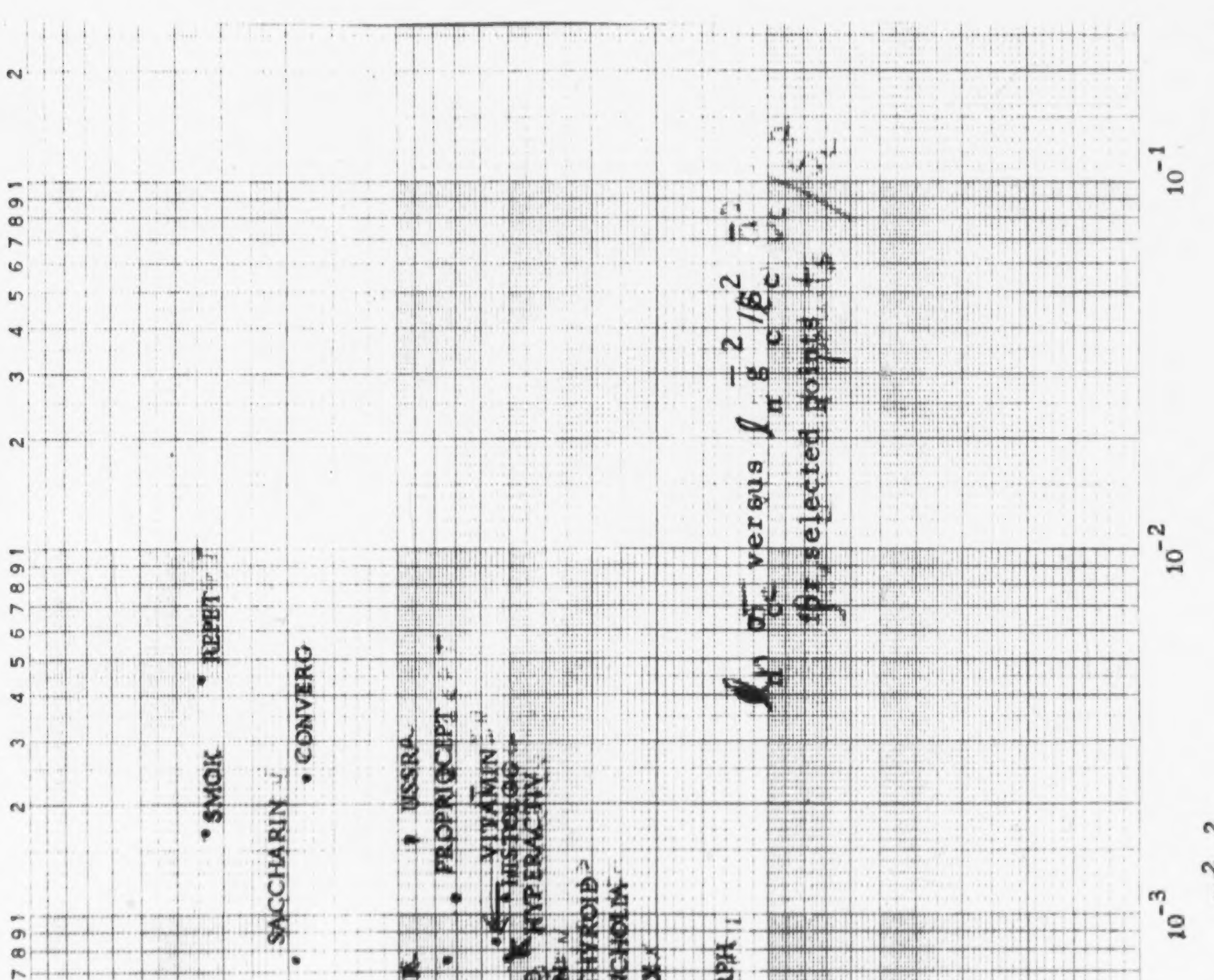


Figure V. 18. Selected Points σ_c Versus $\bar{g}^2/S_c^2$

RANK	STEM	V_c	σ_c	$\bar{g}_c^2/S_c^2$
1	TBR	$.31448 \times 10^6$	10	$.3180 \times 10^{-4}$
2	AZAURACIL	$.29327 \times 10^6$	13	$.4433 \times 10^{-4}$
3	BPW	$.29002 \times 10^6$	8	$.2758 \times 10^{-4}$
29	POSTINSTITU	$.25377 \times 10^6$	7	$.2758 \times 10^{-4}$
77	NAVAHO	$.21704 \times 10^6$	11	$.5068 \times 10^{-4}$
88	SMOK	$.19876 \times 10^6$	342	$.1721 \times 10^{-2}$
109	SR	$.18147 \times 10^6$	98	$.5400 \times 10^{-3}$
195	ELOP	$.17007 \times 10^6$	15	$.8820 \times 10^{-4}$
258	SACCHARIN	$.15706 \times 10^6$	119	$.7577 \times 10^{-3}$
329	AMNESIAC	$.14501 \times 10^6$	4	$.2758 \times 10^{-4}$
672	ANOMIA	$.12958 \times 10^6$	17	$.1312 \times 10^{-3}$
807	BRAND	$.12034 \times 10^6$	46	$.3822 \times 10^{-3}$
1057	MIGRAIN	$.10994 \times 10^6$	36	$.3275 \times 10^{-3}$
1802	JEWISH	$.10062 \times 10^6$	55	$.5466 \times 10^{-3}$
1827	HYPOTHYROID	$.99966 \times 10^5$	33	$.3301 \times 10^{-3}$
2000	ETHER	$.96595 \times 10^5$	72	$.7454 \times 10^{-3}$
3000	APHAGIA	$.83359 \times 10^5$	12	$.1440 \times 10^{-3}$
3885	MELANCHOLIA	$.75799 \times 10^5$	26	$.3430 \times 10^{-3}$
4000	REPET	$.74965 \times 10^5$	340	$.4535 \times 10^{-2}$
4500	CONVERG	$.72094 \times 10^5$	175	$.2427 \times 10^{-2}$
4999	ECTOMORPH	$.69121 \times 10^5$	15	$.2170 \times 10^{-3}$
5502	VITAMIN	$.66626 \times 10^5$	57	$.8555 \times 10^{-3}$
6000	HYPERACTIV	$.64229 \times 10^5$	50	$.7785 \times 10^{-3}$
6501	HEBOID	$.62095 \times 10^5$	5	$.8052 \times 10^{-4}$
7000	PROPRIOCEPT	$.60158 \times 10^5$	69	$.1147 \times 10^{-2}$
7505	USSR	$.58108 \times 10^5$	94	$.1618 \times 10^{-2}$
7571	DYSLEX	$.57920 \times 10^5$	21	$.3626 \times 10^{-3}$
7573	LEUCOCYT	$.57919 \times 10^5$	15	$.2590 \times 10^{-3}$
8499	PHOTOCOAGULA	$.54827 \times 10^5$	3	$.5472 \times 10^{-4}$
9500	HYDROXYZIN	$.51623 \times 10^5$	6	$.1162 \times 10^{-3}$
10502	HISTOLOG	$.48033 \times 10^5$	51	$.1062 \times 10^{-2}$
11006	CORTICOID	$.45868 \times 10^5$	4	$.8721 \times 10^{-4}$
11501	ANTISCIENTIF	$.43691 \times 10^5$	3	$.6866 \times 10^{-4}$
13501	MEGALOMANIAC	$.36385 \times 10^5$	3	$.8245 \times 10^{-4}$

Table ~~V~~. 1

V. 5. 3 Program Description: MRGSF1 and SF2. B

V. 5. 3. 1 Phase I Of The Filter. Phase I of the STATISTICAL FILTER performs three functions: microconcordance merging to build the master concordance, calculation of Dennis measure for each stem in the data base, and dimension reduction of the master concordance.

Input to the first phase of the filter is the sorted microconcordance tape. For each abstract tape processed by the microconcordance a separate output tape of microconcordances is generated. These types are sorted individually on two keys:

1) PRIMARY KEY - alpha-order of stem
2) SECONDARY KEY - descending weight of stem

Finally, all sorted microconcordance tapes are merged on the same keys, thus generating the proper format for input to the first phase of the filter. Also, the document leaders, the total number of items per abstract, are used by Phase I of the filter to calculate the Dennis Measure.

All microconcordances for a particular stem are concatenated sequentially as they appear on the input tape to generate the master concordance, from which Dennis' Measure is calculated. The master concordance is allowed to contain a maximum of 30 DOCUMENT/WEIGHT pairs; if this is exceeded, the DOCUMENT/WEIGHT pair string is truncated, lopping off all low-weight terms.

Two input tapes are generated by Phase I of the STATISTICAL FILTER:

1) An alpha-ordered list of all stems and then Dennis' Measure.
2) The (S_c^2 and S_c^2/V_c are also included for evaluation studies) dimension-reduced master concordance (DOCUMENT/WEIGHT pairs ordered by descending WEIGHT).

Prior to Phase II of the STATISTICAL FILTER the tape of stems and Dennis' Measures is sorted by descending Dennis Measure value. The 5,000 stems with the greatest Dennis Measure are isolated on a separate file and resorted in alpha-order of the stems to generate the final dictionary tape.

V.5.3.2 Phase II of the Filter. Phase II of the STATISTICAL FILTER isolates the master concordance of all dictionary stems. Briefly, a table of dictionary stems is generated from the dictionary tape. In turn, each master concordance is read, and the stem checked against the stem dictionary by means of a binary search algorithm. If the stem is indeed present in the dictionary, the DOCUMENT/WEIGHT pairs of that master concordance are re-ordered by ascending document number and the master concordance is written onto the output tape of Dictionary Stem Master Concordances which serves as input to Phase I of the Stem Clustering algorithm, routine CLUSTR.0.

V.5.3.3 Flowcharts. The following discussion is intended to describe this phase of the filter, called MRGSF1, in detail. Flowchart comments:

The following are labelled explanations of program MRGSF1 flowchart (see Figure V. 19).

1) Open input and output files
2) Build a table of floating point numbers from 0.0 to 249.0 for use in integer to floating point conversion
3) Read file LEADER, the tape of document leaders
4) EOF LEADER? Yes = 6, No = 5
5) Fill table LD with the document stem count leaders; sequential table position implicitly defines document number. This table contains the stem count for every abstract in the collection.
6) Read file RAWCON, the microconcordance tape
7) EOF RAWCON? Yes = 8; No = 9
8) If RAWCON is EOF, close all input and output files and terminate job. An EOF should not be obtained at this point because this I/O statement is used only once to read the

first microconcordance on tape. The EOF check is made only as a precautionary check.

9) This code initializes the building of the master concordance. The stem is entered into words 2 and 3, while the first document number and weight pair is packed and entered into word 4. Word 1 is left vacant to be filled with the D/W pair count when the master concordance D/W pair string has been completed.

10) Read a microconcordance from file RAWCON (see item 6)

11) EOF RAWCON? Yes = 12; No = 15

12) Calculate number of D/W pairs and number of words in master concordance logical record. This is done by arithmetic manipulation of the table base address stored in index register 6 and the last entry address stored in index register 5.

13) Transfer to CLOSE SF1, the subroutine which calculates the Dennis Measure

14) Close all I/O files and terminate the job

15) Test for a new stem in the input buffer

16) If it is a new stem, perform the same function as described in Item 12

17) Transfer to CLOSE SF1 (see Item 13)

18) Pack the document number and stem count and store in concordance buffer. Increment D/W pair overflow tally counter and increment address stored in Index Register 5.

19) Check for D/W pair buffer overflow. If overflow is detected, the master concordance is deleted.

20) D/W buffer overflow has been reached. Read file RAWCON

21) If an EOF is detected, go immediately to the wrapup routine since the master concordance now residing in the buffer is to be deleted.

22) Check for the occurrence of a new stem in the microconcordance input buffer.

23) CLOSE SF1 (the statistical filter routine).

24) Fetch the D/W pair count from the master concordance buffer.

25) Using a modified D. L. SHELL sinking sort, order the D/W pairs of the master concordance in order of descending stem weights.

26) Check for a D/W pair count greater than 30. This is a test for dimension-reduction.

27) The D/W pair count is greater than 30. The master concordance D/W pair count is changed to 30. (Note: although the master concordance D/W pair string is limited to 30 pairs, all pairs are used in the calculation of the Dennis measure.)

28) Write a master concordance of 30 D/W pairs onto tape (file name RESORT).

29) Write a master concordance of less than 30 D/W pairs onto tape (file RESORT).

30) Initialize all variables and constants used in the calculation of the Dennis measure.

31) The three constants ALPHAC, BETAC, and SIGMAC are calculated within this loop. The loop is reiterated until the master concordance D/W string is exhausted (see item 27).

32) Load the Q-register with the loop index which is then used as an address modifier to load the appropriate D/W pair from the master concordance into the A-register; in turn, the lower A-register which contains the stem weight is used as an address modifier to fetch the equivalent floating point number from a conversion table. This floating point number is stored in item WEIGHT.

33) Load the Q-register with the loop index which is then used as an address modifier to load the appropriate D/W pair from the master concordance into the A-register; the upper A-register which contains the document number is used as an address modifier to fetch the corresponding document leader from table LD (see item 5). An integer to floating point conversion is performed as in item 32; the floating-pointer leader count is then stored in item TOTSTM.

34) SIGMAC is calculated; the mathematical expression is as follows:

$$\text{SIGMAC} = \sum_{d=1}^{D} (f_{c,d})$$

where:

D = number of documents in the collection

$f_{c,d}$ = number of occurrences of stem c in document d.

The master concordance does not contain a D/W pair for every document in the collection for obvious reasons; the inclusion of D/W pairs where $f_c = 0$ in a summation series of $f_{c,d}$ is inefficient and superfluous. Also, the inclusion of such terms would make the array dimensions prohibitive.

35) BETAC is calculated; the mathematical expression is as follows:

$$\text{BETAC} = \sum_{d=1}^{D} (g_{c,d})$$

where:

$$g_{c,d} = f_{c,d}/Ld$$

Ld = number of stems in document d (the document leader).

36) ALPHAC is calculated; the mathematical expression is as follows:

$$\text{ALPHAC} = \sum_{d=1}^{D} (g_{c,d})^2$$

(See Item 35 for the definition of $g_{c,d}$.)

37) "End-of-loop?", return or exit.

38) Calculate the Dennis Measure

39) Build the STEM/DENNIS MEASURE pair (item STEMVC)

40) Write the STEM/DENNIS MEASURE pair onto magnetic tape (file STVC)

41) Return to the main program from CLOSE SF1.

V.5.3.4 Flowchart Comments for Routine SF2.B, Phase II of the Statistical Filter. (refers to Figure V.20).

1) Open input and output files.

2) Preset item RANGE, the number of dictionary stems, to the value defined as DICT.

3) Fill table CONSTM with the dictionary stems (2 words per stem). If the number of stems input is less than the predefined size of the dictionary file, reset item RANGE to the proper value.

4) Read a master concordance from input file RESORT into table PAIRS.

5) EOF RESORT? Yes ⇒7, No ⇒6

6) Isolate the tally of D/W pairs for this master concordance and store in item PRCNT. Isolate the stem and store in table STORE.

7) Close all input and output files. Terminate job!!!

8) Perform a binary search in order to determine if the current master concordance stem is a dictionary stem.

9) Was the stem found in the dictionary? Yes ⇒9, No ⇒4

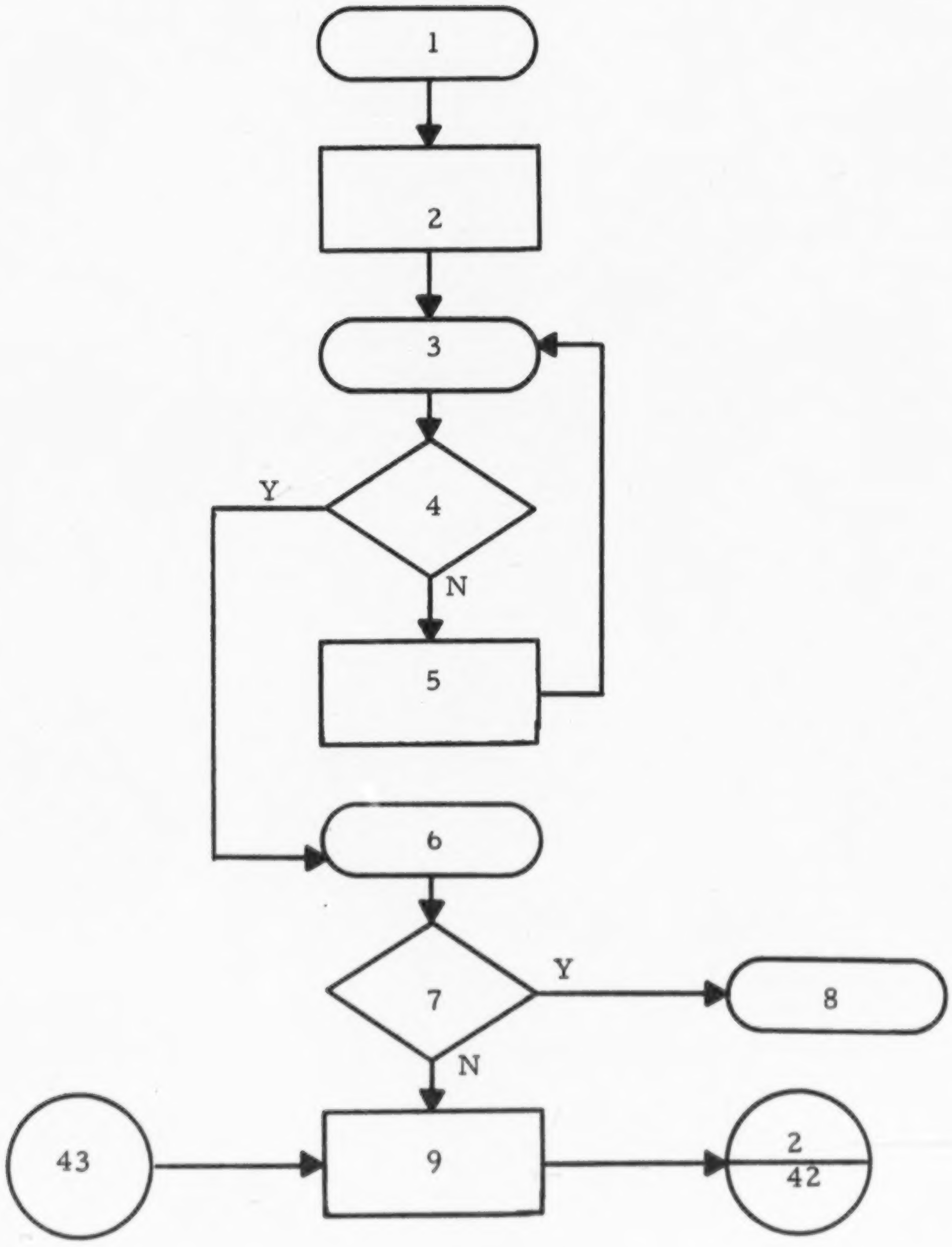

1
2
3
Y
4
N
5
6
7
Y
8
N
43
9
2
42

Figure V.19. MRGSF1 Flowchart

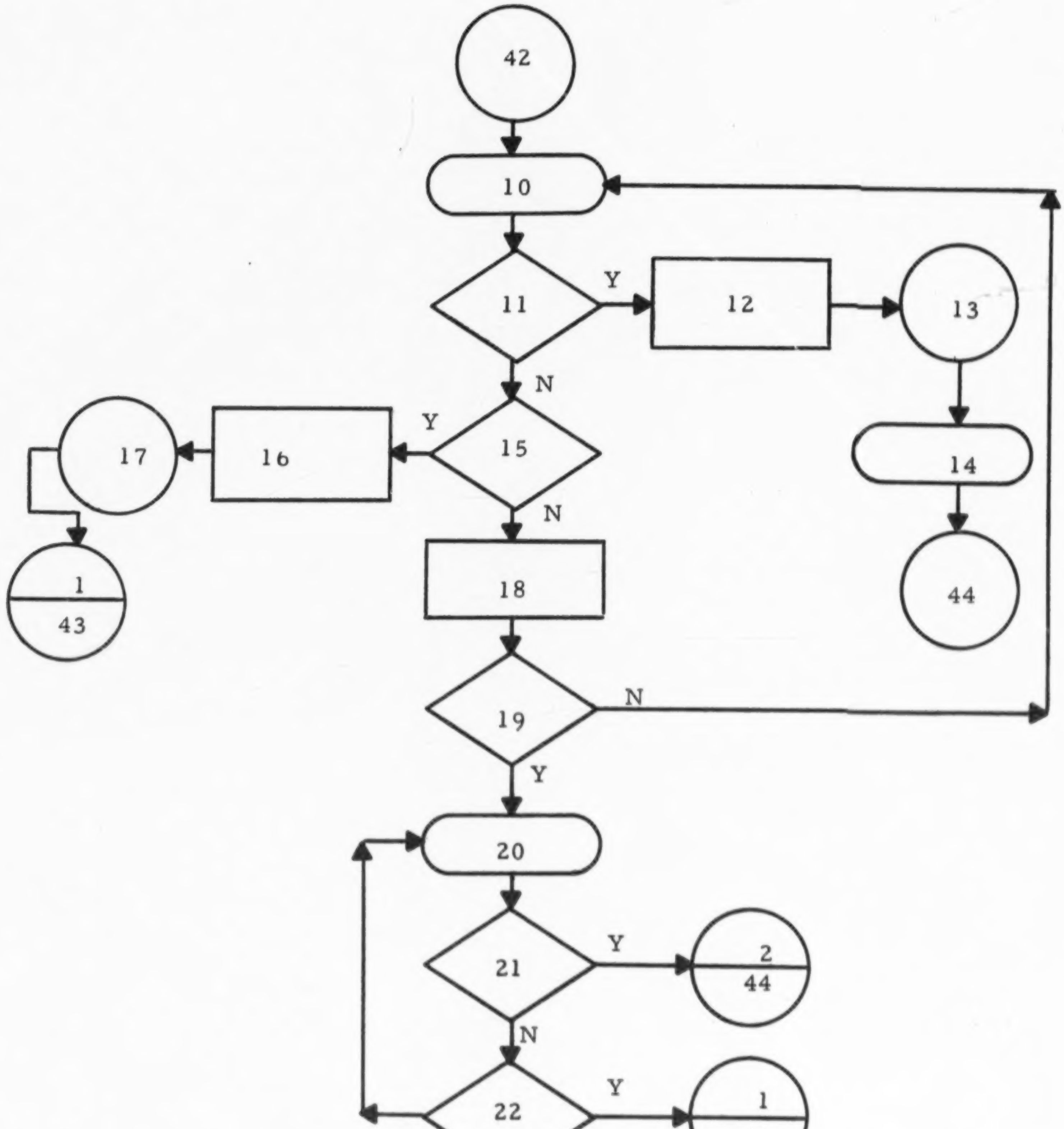

42
10
11
Y
12
13
N
15
Y
16
17
1
43
N
14
44
18
19
N
Y
20
21
Y
2
44
N
22
Y
1
43

Figure V.19. (Continued)

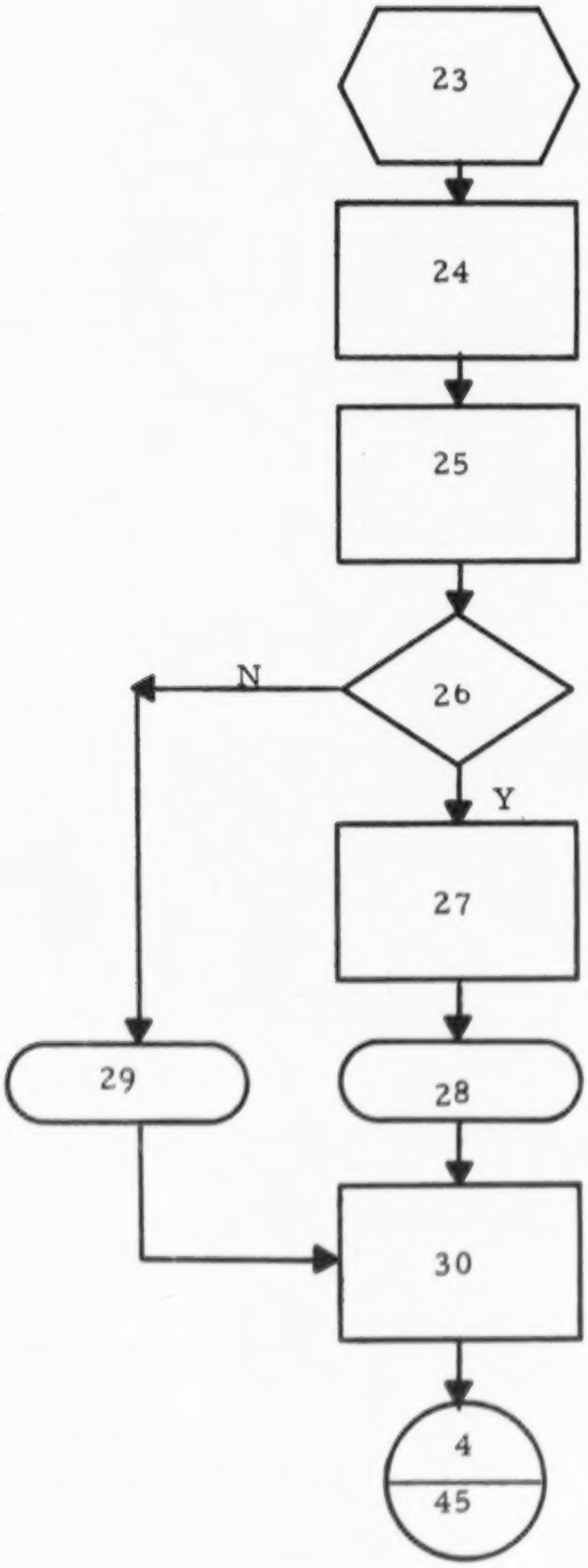

23
24
25
N
26
Y
27
29
28
30
4
45

Figure V. 19. (Continued)

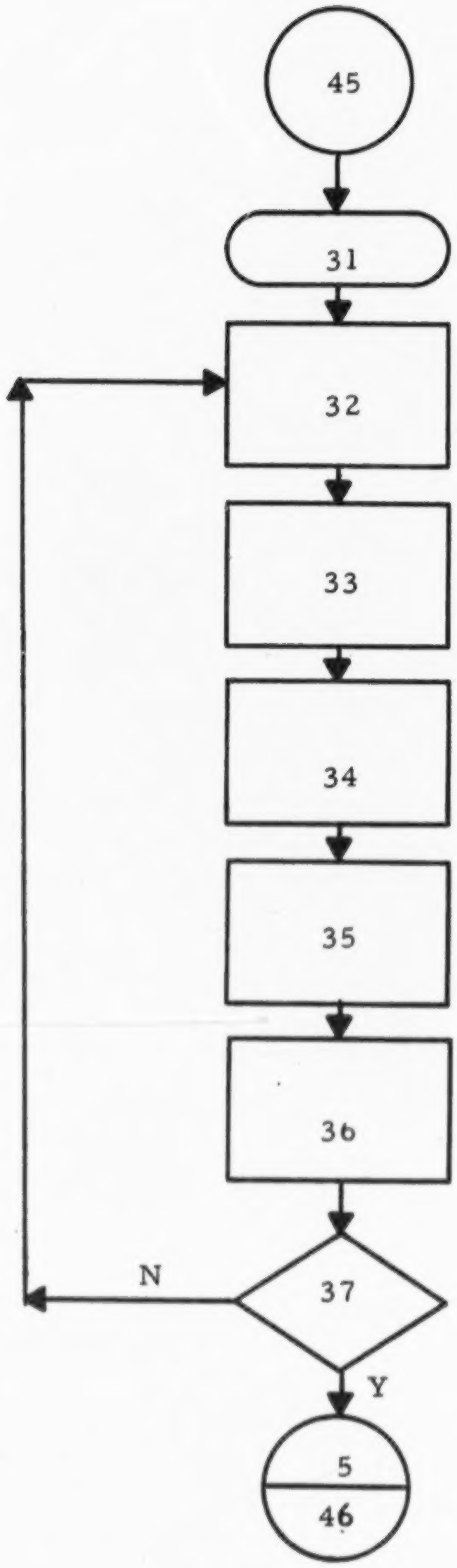

45
31
32
33
34
35
36
37
N
Y
5
46

Figure V. 19. (Continued)

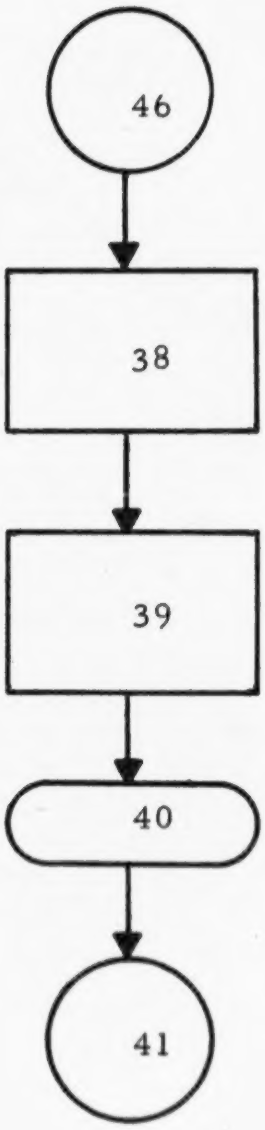
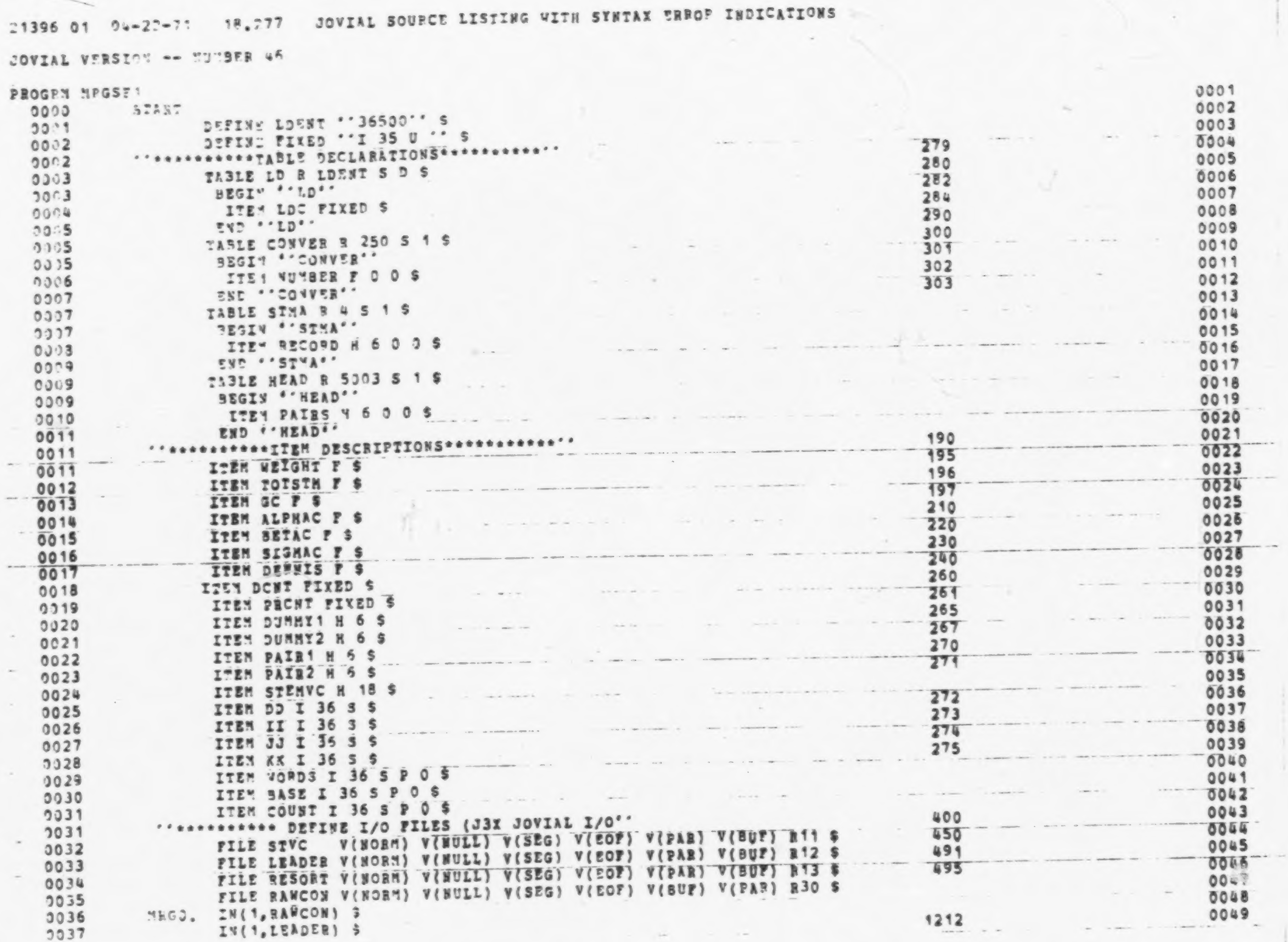

46
38
39
40
41

Figure V. 19. (Concluded)



```
21396 01  04-22-71   18.277   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM MRGSF1                                                                          0001
  0000       START                                                                     0002
  0001            DEFINE LDENT ''36500'' $                                             0003
  0002            DEFINE FIXED ''I 35 U '' $                               279         0004
  0002     ''***********TABLE DECLARATIONS***********''                    280         0005
  0003            TABLE LD R LDENT S D $                                   282         0006
  0003             BEGIN ''LD''                                            284         0007
  0004              ITEM LDC FIXED $                                       290         0008
  0005             END ''LD''                                              300         0009
  0005            TABLE CONVER R 250 S 1 $                                 301         0010
  0005             BEGIN ''CONVER''                                        302         0011
  0006              ITEM NUMBER F 0 0 $                                    303         0012
  0007             END ''CONVER''                                                      0013
  0007            TABLE STMA R 4 S 1 $                                                 0014
  0007             BEGIN ''STMA''                                                      0015
  0008              ITEM RECORD H 6 0 0 $                                              0016
  0009             END ''STMA''                                                        0017
  0009            TABLE HEAD R 5003 S 1 $                                              0018
  0009             BEGIN ''HEAD''                                                      0019
  0010              ITEM PAIRS H 6 0 0 $                                               0020
  0011             END ''HEAD''                                            190         0021
  0011      ''***********ITEM DESCRIPTIONS***********''                    195         0022
  0011             ITEM WEIGHT F $                                         196         0023
  0012             ITEM TOTSTM F $                                         197         0024
  0013             ITEM GC F $                                             210         0025
  0014             ITEM ALPHAC F $                                         220         0026
  0015             ITEM BETAC F $                                          230         0027
  0016             ITEM SIGMAC F $                                         240         0028
  0017             ITEM DENNIS F $                                         260         0029
  0018            ITEM DCNT FIXED $                                        261         0030
  0019             ITEM PRCNT FIXED $                                      265         0031
  0020             ITEM DUMMY1 H 6 $                                       267         0032
  0021             ITEM DUMMY2 H 6 $                                       270         0033
  0022             ITEM PAIR1 H 6 $                                        271         0034
  0023             ITEM PAIR2 H 6 $                                                    0035
  0024             ITEM STEMVC H 18 $                                      272         0036
  0025             ITEM DD I 36 S $                                        273         0037
  0026             ITEM II I 36 S $                                        274         0038
  0027             ITEM JJ I 36 S $                                        275         0039
  0028             ITEM KK I 36 S $                                                    0040
  0029             ITEM WORDS I 36 S P 0 $                                             0041
  0030             ITEM BASE I 36 S P 0 $                                              0042
  0031             ITEM COUNT I 36 S P 0 $                                 400         0043
  0031       ''*********** DEFINE I/O FILES (J3X JOVIAL I/O''              450         0044
  0032             FILE STVC   V(NORM) V(NULL) V(SEG) V(EOF) V(PAR) V(BUF) R11 $   491         0045
  0033             FILE LEADER V(NORM) V(NULL) V(SEG) V(EOF) V(PAR) V(BUF) R12 $   495         0046
  0034             FILE RESORT V(NORM) V(NULL) V(SEG) V(EOF) V(PAR) V(BUF) R13 $               0047
  0035             FILE RAWCON V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R30 $               0048
  0036      MRGO.  IN(1,RAWCON) $                                          1212        0049
  0037             IN(1,LEADER) $
```

Figure V. 19a. MRGSF1 Listing

```
21396 01  04-22-71   18.277   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46
                                                                    1214               0050
 0038             OUT(1,STVC) $                                     1216               0051
 0039             OUT(1,RESORT) $                                   1217               0052
 0040             ALPHAC=0.0 $                                      1218               0053
 0041             FOR N=0,1,249 $                                   1220               0054
 0042              BEGIN ''N''                                      1224               0055
 0043               NUMBER($N$)=ALPHAC $                            1226               0056
 0044               ALPHAC=ALPHAC+1.0 $                             1228               0057
 0045              END ''N''                                        1230               0058
 0046             DIRECT                                            1240               0059
 0047             STZ     DCNT                                      1250               0060
 0048             JOVIAL                                                               0061
 0049     LOAD.   IO10(0,LEADER,0,LDC($DCNT$),1) $                  1252               0062
 0050             WAIT(IO1) $                                       1253               0063
 0051             IF LEADER EQ V(EOF) $                                                0064
 0052             GOTO MRG6 $                                       1255               0065
 0053             DIRECT                                            1256               0066
 0054             AOS     DCNT                                      1257               0067
 0055             JOVIAL                                            1259               0068
 0056             GOTO LOAD $                                                          0069
 0057     MRG6.   IO1(0,RAWCON,0,STMA,4) $                                             0070
 0058             WAIT(IO1) $                                                          0071
 0059             IF RAWCON EQ V(EOF) $                                                0072
 0060             GOTO MRG5 $                                                          0073
 0061             DIRECT                                                               0074
 0062     MRG1    EAX6    HEAD                                                         0075
 0063             EAX5    HEAD+2                                                       0076
 0064             STZ     WORDS                                                        0077
 0065             STZ     BASE                                                         0078
 0066             STZ     COUNT                                                        0079
 0067             LDAQ    STMA+1                                                       0080
 0068             STA     0,5                                                          0081
 0069             STQ     1,5                                                          0082
 0070             ADX5    2,DU                                                         0083
 0071             LDAQ    STMA+3                                                       0084
 0072             QLR     18                                                           0085
 0073             LLR     18                                                           0086
 0074             STA     0,5                                                          0087
 0075             JOVIAL                                                               0088
 0076     MRG2.   IO2(0,RAWCON,0,STMA,4) $                                             0089
 0077             WAIT(IO2) $                                                          0090
 0078             IF RAWCON EQ V(EOF) $                                                0091
 0079             GOTO MRG4 $                                                          0092
 0080             DIRECT                                                               0093
 0081             LDAQ    STMA+1                                                       0094
 0082             CMPA    HEAD+2                                                       0095
 0083             TNZ     MRG3                                                         0096
 0084             CMPQ    HEAD+3                                                       0097
 0085             TNZ     MRG3                                                         0098
 0086             LDAQ    STMA+3                                                       0099
 0087             QLR     18
```

Figure V. 19a. (Continued)

```
21396 01  04-22-71   18,277   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0088              LLR      18                                                     0100
  0089              ADX5     1,DU                                                   0101
  0090              AOS      COUNT                                                  0102
  0091              STA      0,5                                                    0103
  0092              LDA      COUNT                                                  0104
  0093              SBA      5000,DL                                                0105
  0094              TMI      MRG2                                                   0106
  0095              JOVIAL                                                          0107
  0096      INITO.  IO54(0,RAWCON,0,STMA,4) $                                       0108
  0097              WAIT(IO54) $                                                    0109
  0098              IF RAWCON EQ V(EOF) $                                           0110
  0099              GOTO MRG5 $                                                     0111
  0100              DIRECT                                                          0112
  0101              LDAQ     STMA+1                                                 0113
  0102              CMPA     HEAD+2                                                 0114
  0103              TNZ      MRG1                                                   0115
  0104              CMPQ     HEAD+3                                                 0116
  0105              TNZ      MRG1                                                   0117
  0106              TRA      INITO                                                  0118
  0107      MRG3    STX6     BASE                                                   0119
  0108              SBX5     BASE                                                   0120
  0109              SXL5     WORDS                                                  0121
  0110              SBX5     3,DU                                                   0122
  0111              SXL5     1,6                                                    0123
  0112              JOVIAL                                                          0124
  0113              GOTO SP1 $                                                      0125
  0114              GOTO MRG1 $                                                     0126
  0115              DIRECT                                                          0127
  0116      MRG4    STX6     BASE                                                   0128
  0117              SBX5     BASE                                                   0129
  0118              SXL5     WORDS                                                  0130
  0119              SBX5     3,DU                                                   0131
  0120              SXL5     1,6                                                    0132
  0121              JOVIAL                                                          0133
  0122              GOTO SP1 $                                                      0134
  0123      MRG5.   IN(2,RAWCON) $                                                  0135
  0124              IN(2,LEADER) $                                      1780        0136
  0125              OUT(2,STVC) $                                       1800        0137
  0126              OUT(2,RESORT) $                                     1790        0138
  0126      ''*****************************************************''               0139
  0126      ''                         SP1.B                          ''            0140
  0126      ''*****************************************************''               0141
  0127            CLOSE SP1 $                                                       0142
  0128             BEGIN ''SP1.B''                                                  0143
  0129              DIRECT                                              1350        0144
  0130              LDA      HEAD+1                                                 0145
  0131              STA      PRCNT                                      1360        0146
  0132              JOVIAL                                              1365        0147
  0133      SRT0.     LL=2 $                                             710        0148
  0134      SRT1.     IF LL GO PRCNT $                                   720        0149
```

Figure V. 19a. (Continued)

```
21396 01  04-22-71   18,277   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0135               GOTO SRT2 $                                  730                 0150
 0136               LL=LL*2 $                                    740                 0151
 0137               GOTO SRT1 $                                  750                 0152
 0138       SRT2.   DD=(LL/2)-1 $                                760                 0153
 0141       SRT3.   II=0 $  JJ=0 $  KK=0 $                       770                 0154
 0142               DIRECT                                       780                 0155
 0143       SRT4   LDA     II                                    790                 0156
 0144              EAX7    HEAD+4,AL                                                 0157
 0145              LDQ     0,7                                   810                 0158
 0146              STCQ    PAIR1,07                              820                 0159
 0147              ADA     DD                                    830                 0160
 0148              EAX6    HEAD+4,AL                                                 0161
 0149              LDQ     0,6                                   850                 0162
 0150              STCQ    PAIR2,07                              860                 0163
 0151              LDA     PAIR1                                 870                 0164
 0152              CMPA    PAIR2                                 880                 0165
 0153              TMI     SRT7                                  890                 0166
 0154               JOVIAL                                       900                 0167
 0155       SRT5.   IF II+DD+1 LS PRCNT $                                            0168
 0156                BEGIN                                       920                 0169
 0157                 II=II+1 $                                                      0170
 0158                 GOTO SRT4 $                                940                 0171
 0159                END                                         950                 0172
 0160               DD=(DD-1)/2 $                                960                 0173
 0161               IF DD LQ 0 $                                                     0174
 0162               GOTO OPTN1 $                                                     0175
 0163               GOTO SRT3 $                                  980                 0176
 0164               DIRECT                                       990                 0177
 0165       SRT7   LDA     0,7                                   1000                0178
 0166              STA     DUMMY1                                1002                0179
 0167              LDA     0,6                                   1004                0180
 0168              STA     0,7                                   1006                0181
 0169              LDA     DUMMY1                                1008                0182
 0170              STA     0,6                                   1010                0183
 0171               JOVIAL                                       1012                0184
 0172               JJ=II-DD $                                   1014                0185
 0173               IF JJ LS 0 $                                 1016                0186
 0174               GOTO SRT5 $                                  1018                0187
 0175               KK=II $                                      1020                0188
 0176               DIRECT                                       1022                0189
 0177       SRT8   LDA     JJ                                    1024                0190
 0178              EAX7    HEAD+4,AL                                                 0191
 0179              LDQ     0,7                                   1028                0192
 0180              STCQ    PAIR1,07                              1030                0193
 0181              LDA     KK                                    1032                0194
 0182              EAX6    HEAD+4,AL                                                 0195
 0183              LDQ     0,6                                   1035                0196
 0184              STCQ    PAIR2,07                              1036                0197
 0185              LDA     PAIR1                                 1038                0198
 0186              CMPA    PAIR2                                 1040                0199
```

Figure V. 19a. (Continued)

```
21396 01  04-22-71   18.277   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS
JOVIAL VERSION -- NUMBER 46

 0187              TZE     SRT5                                  1042          0200
 0188              TPL     SRT5                                  1044          0201
 0189              LDA     0,7                                   1046          0202
 0190              STA     DUMMY1                                1048          0203
 0191              LDA     0,6                                   1050          0204
 0192              STA     0,7                                   1052          0205
 0193              LDA     DUMMY1                                1054          0206
 0194              STA     0,6                                   1056          0207
 0195                JOVIAL                                      1058.         0208
 0196                JJ=JJ-DD $                                  1060          0209
 0197                IF JJ LS 0 $                                1062          0210
 0198                GOTO SRT5 $                                 1064          0211
 0199                KK=JJ+DD $                                  1066          0212
 0200                GOTO SRT8 $                                 1068          0213
 0201        OPTN1. IF PRCNT GR 30 $                                           0214
 0202               BEGIN                                                      0215
 0203                DIRECT                                                    0216
 0204              LDA     30,DL                                               0217
 0205              STA     HEAD+1                                              0218
 0206                JOVIAL                                                    0219
 0207                IO17(4,RESORT,0,HEAD,33) $                                0220
 0208                WAIT(IO17) $                                              0221
 0209                GOTO CALC $                                               0222
 0210               END                                                        0223
 0211              IO16(4,RESORT,0,HEAD,PRCNT+3) $                             0224
 0212              WAIT(IO16) $                                                0225
 0213        CALC.  ALPHAC=0.0 $                                               0226
 0214              BETAC=0.0 $                                   1442          0227
 0215              SIGMAC=0.0 $                                  1444          0228
 0216              GC=0.0 $                                      1446          0229
 0217              WEIGHT=0.0 $                                  1447          0230
 0218              TOTSTM=0.0 $                                  1448          0231
 0219              DENNIS=0.0 $                                  1449          0232
 0220              FOR I=0,1,PRCNT-1 $                           1450          0233
 0221               BEGIN ''I''                                  1460          0234
 0222              DIRECT                                        1470          0235
 0223              LDQ      I                                    1472          0236
 0224              LDA      HEAD+4,QL                                          0237
 0225              LDQ      CONVER+1,AL                          1478          0238
 0226              STQ      WEIGHT                               1480          0239
 0227              LDQ      I                                    1482          0240
 0228              LDA      HEAD+4,QL                                          0241
 0229              LDQ      LD,AU                                1488          0242
 0230              LDA      CONVER+1,QL                          1490          0243
 0231              STA      TOTSTM                               1492          0244
 0232              FLD      WEIGHT                               1494          0245
 0233              FAD      SIGMAC                               1496          0246
 0234              FST      SIGMAC                               1498          0247
 0235              FLD      WEIGHT                               1500          0248
 0236              FDV      TOTSTM                               1502          0249
```

Figure V. 19a. (Continued)

```
21396 01  04-22-71    18.277   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0237           FST     GC
 0238           FLD     GC                                                1504              0250
 0239           FAD     BETAC                                             1506              0251
 0240           FST     BETAC                                             1508              0252
 0241           FLD     GC                                                1510              0253
 0242           FMP     GC                                                1512              0254
 0243           FAD     ALPHAC                                            1514              0255
 0244           FST     ALPHAC                                            1516              0256
 0245           JOVIAL                                                    1518              0257
 0246            END ''I''                                                1520              0258
 0247           BETAC=BETAC*BETAC $                                       1530              0259
 0248           DENNIS=SIGMAC*(((ALPHAC*DCNT)/BETAC)-1.0) $               1540              0260
 0249           DIRECT                                                    1700              0261
 0250           LDA     HEAD+2                                            1720              0262
 0251           STA     STEMVC                                                              0263
 0252           LDA     HEAD+3                                            1722              0264
 0253           STA     STEMVC+1                                                            0265
 0254           LDA     DENNIS                                            1726              0266
 0255           STA     STEMVC+2                                          1727              0267
 0256           JOVIAL                                                    1728              0268
 0257           IO9(4,STVC,0,STEMVC,3) $                                  1735              0269
 0258           WAIT(IO9) $                                                                 0270
 0259           RETURN $                                                  1750              0271
 0260          END ''SF1,B''                                                                0272
 0261           STOP $                                                                      0273
 0262           TERM MBGO $                                                                 0274
                                                                                            0275
```

Figure V. 19a. (Concluded)

10) Resort the D/W pairs of the current master concordance by means of a modified D. L. SHELL sinking-sort routine.

11) Write the sorted master concordance onto output file REDUCE.

12) Increment item CNTOUT, the counter of master concordance found in the dictionary and subsequently written on the output file.

13) If CNTOUT is equal to RANGE, all dictionary concordances have been found. If so, terminate job!!!

CNTOUT = RANGE? Yes ⇒7, No ⇒4

V.5.4 Run Times and Process Chart for the Statistical Filter

The following is a summary of the running times for the statistical filter programs:

Program	CPU Time (In Hours)
MRGSF1	0.4350
SF2.B	0.0188

Figure V.21 illustrates the production process chart for the statistical filter programs, showing the logical flow of the data tapes through the system, including also the actual tape numbers of the different files.

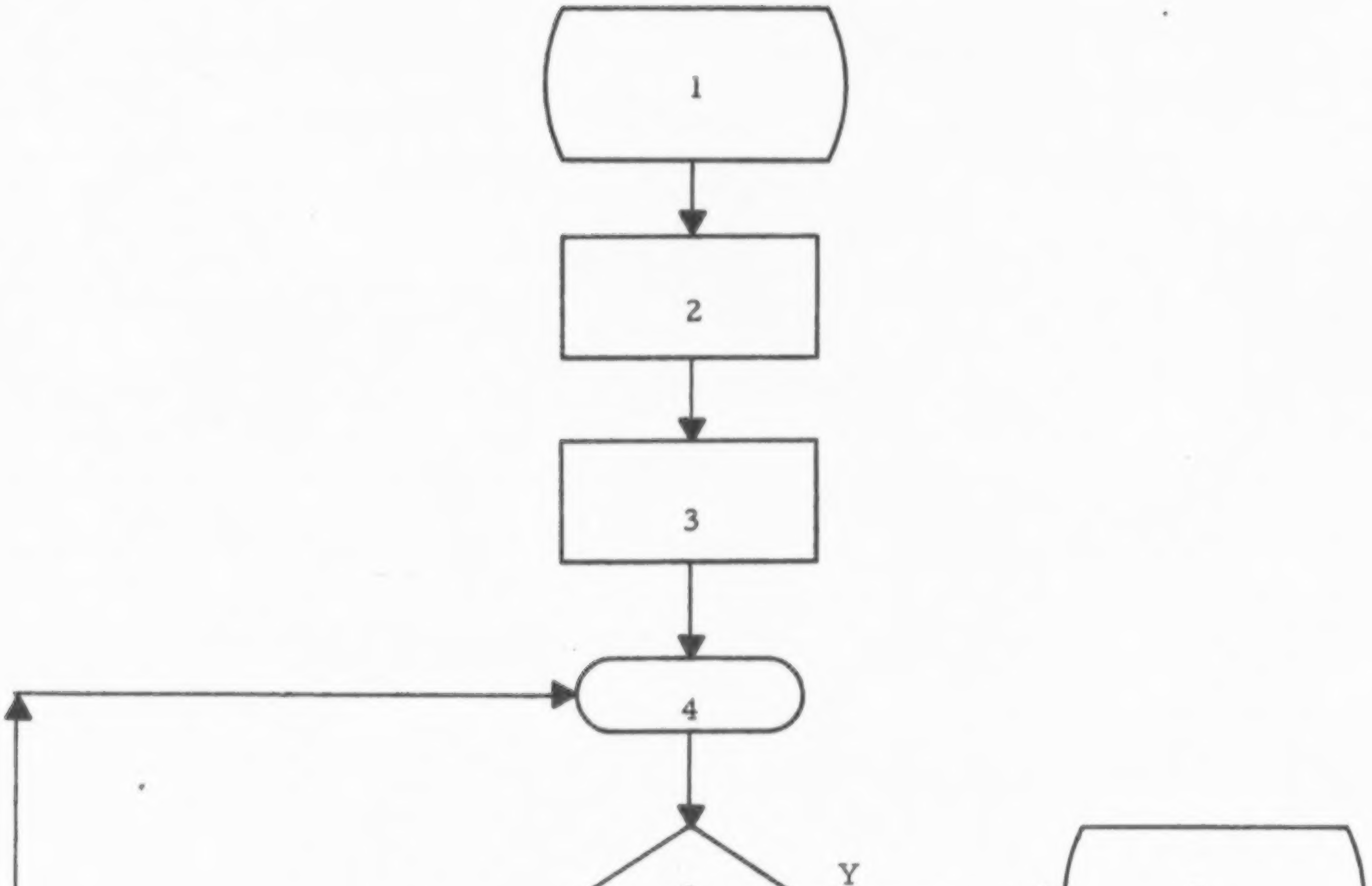

1
2
3
4
5
Y
7
N
6
8
N
9
Y
2
10

Figure V. 20. SF2.B Flowchart

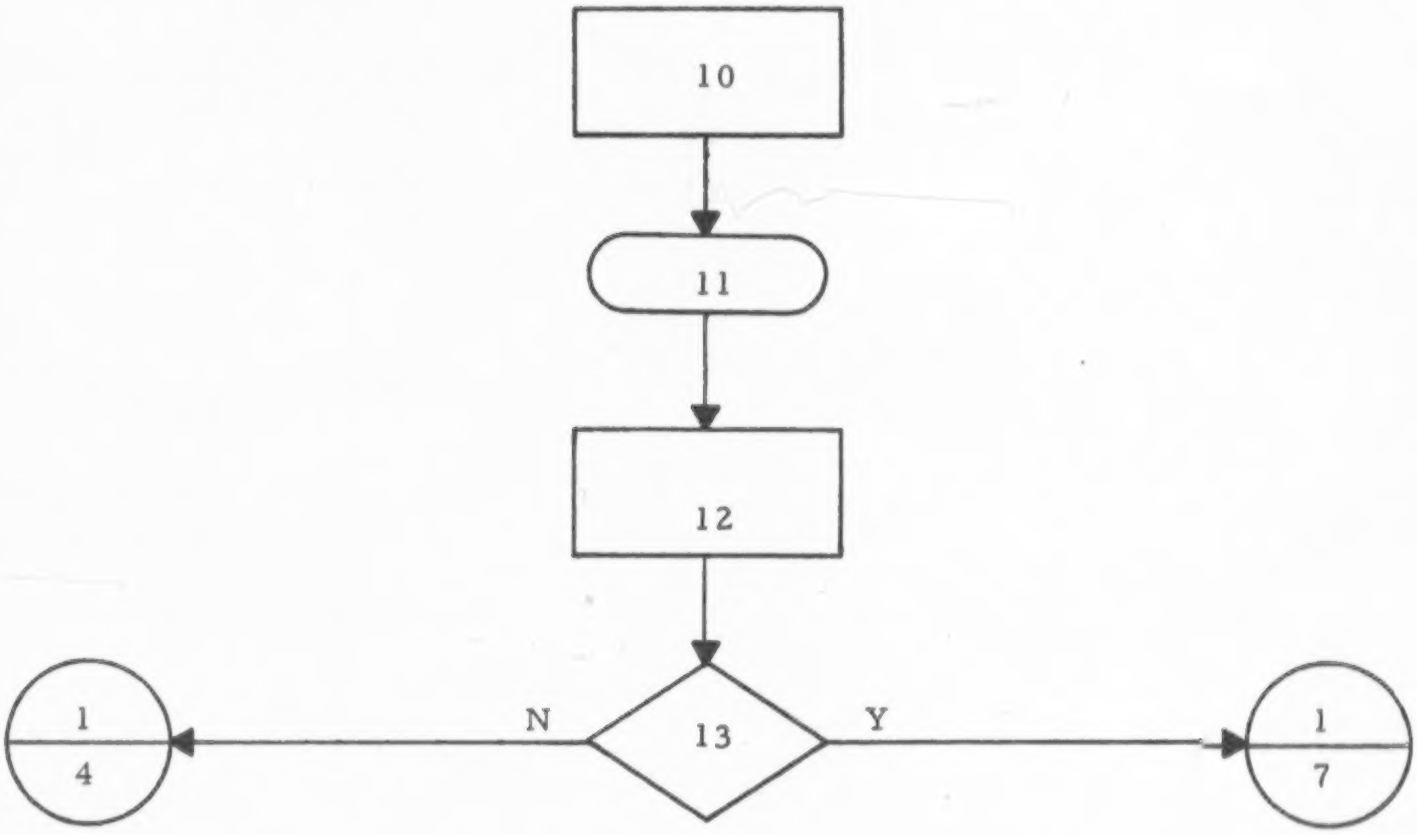
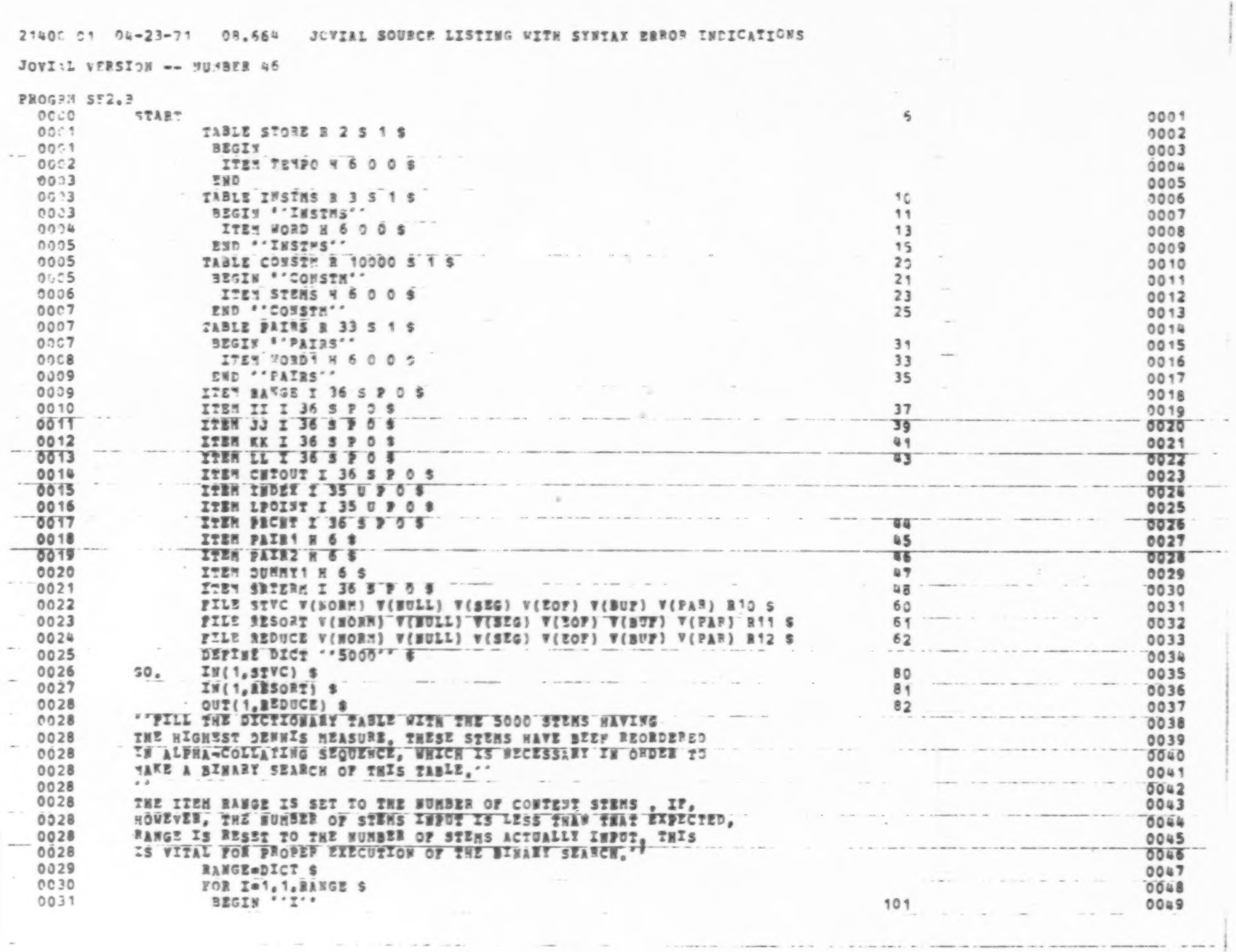

10
11
12
13
N
Y
1
4
1
7

Figure V. 20. (Concluded)



```
21400 01  04-23-71   08.664   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRAM SF2.B
  0000       START                                                            5                        0001
  0001              TABLE STORE R 2 S 1 $                                                              0002
  0001               BEGIN                                                                             0003
  0002                ITEM TEMPO H 6 0 0 $                                                             0004
  0003               END                                                                               0005
  0003              TABLE INSTMS R 3 S 1 $                                                   10        0006
  0003               BEGIN ''INSTMS''                                                        11        0007
  0004                ITEM WORD H 6 0 0 $                                                    13        0008
  0005               END ''INSTMS''                                                          15        0009
  0005              TABLE CONSTM R 10000 S 1 $                                               20        0010
  0005               BEGIN ''CONSTM''                                                        21        0011
  0006                ITEM STEMS H 6 0 0 $                                                   23        0012
  0007               END ''CONSTM''                                                          25        0013
  0007              TABLE PAIRS R 33 S 1 $                                                             0014
  0007               BEGIN ''PAIRS''                                                         31        0015
  0008                ITEM WORD1 H 6 0 0 $                                                   33        0016
  0009               END ''PAIRS''                                                           35        0017
  0009              ITEM RANGE I 36 S P 0 $                                                            0018
  0010              ITEM II I 36 S P 0 $                                                     37        0019
  0011              ITEM JJ I 36 S P 0 $                                                     39        0020
  0012              ITEM KK I 36 S P 0 $                                                     41        0021
  0013              ITEM LL I 36 S P 0 $                                                     43        0022
  0014              ITEM CNTOUT I 36 S P 0 $                                                           0023
  0015              ITEM INDEX I 35 U P 0 $                                                            0024
  0016              ITEM LPOINT I 35 U P 0 $                                                           0025
  0017              ITEM PRCNT I 36 S P 0 $                                                  44        0026
  0018              ITEM PAIR1 H 6 $                                                         45        0027
  0019              ITEM PAIR2 H 6 $                                                         46        0028
  0020              ITEM DUMMY1 H 6 $                                                        47        0029
  0021              ITEM SRTERM I 36 S P 0 $                                                 48        0030
  0022              FILE STVC V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R10 $              60        0031
  0023              FILE RESORT V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R11 $            61        0032
  0024              FILE REDUCE V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R12 $            62        0033
  0025              DEFINE DICT ''5000'' $                                                             0034
  0026       GO.    IN(1,STVC) $                                                             80        0035
  0027              IN(1,RESORT) $                                                           81        0036
  0028              OUT(1,REDUCE) $                                                          82        0037
  0028       ''FILL THE DICTIONARY TABLE WITH THE 5000 STEMS HAVING                                    0038
  0028       THE HIGHEST DENNIS MEASURE, THESE STEMS HAVE BEEN REORDERED                               0039
  0028       IN ALPHA-COLLATING SEQUENCE, WHICH IS NECESSARY IN ORDER TO                               0040
  0028       MAKE A BINARY SEARCH OF THIS TABLE,''                                                     0041
  0028       ''                                                                                        0042
  0028       THE ITEM RANGE IS SET TO THE NUMBER OF CONTENT STEMS , IF,                                0043
  0028       HOWEVER, THE NUMBER OF STEMS INPUT IS LESS THAN THAT EXPECTED,                            0044
  0028       RANGE IS RESET TO THE NUMBER OF STEMS ACTUALLY INPUT, THIS                                0045
  0028       IS VITAL FOR PROPER EXECUTION OF THE BINARY SEARCH.''                                     0046
  0029              RANGE=DICT $                                                                       0047
  0030              FOR I=1,1,RANGE $                                                                  0048
  0031               BEGIN ''I''                                                            101        0049
```

Figure V.20a. SF2.B Listing

```
21400 01  04-23-71   08.664   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0032                IO001(0,STVC,0,INSTMS,2) $                                                                0050
  0033                IF STVC EQ V(EOF) $                                                                       0051
  0034                 BEGIN                                                                                    0052
  0035                  RANGE=I-1 $                                                                             0053
  0036                  GOTO STEP1 $                                                                            0054
  0037                 END                                                                                      0055
  0038                DIRECT                                                                  120               0056
  0039              LDAQ     INSTMS+1                                                         125               0057
  0040        LOCO  STAQ     CONSTM+1                                                         130               0058
  0041              LDA      2,DU                                                             135               0059
  0042              ASA      LOCO                                                             140               0060
  0043                JOVIAL                                                                  145               0061
  0044               END ''I''                                                                150               0062
  0045        STEP1. IO002(0,RESORT,0,PAIRS,33) $                                             200               0063
  0046              IF RESORT EQ V(EOF) $                                                     205               0064
  0047              GOTO IEOF $                                                               210               0065
  0048              DIRECT                                                                                      0066
  0049              LDA      PAIRS+1                                                                            0067
  0050              STA      PRCNT                                                                              0068
  0051              LDA      PAIRS+2                                                                            0069
  0052              LDQ      PAIRS+3                                                                            0070
  0053              STAQ     STORE+1                                                                            0071
  0054              JOVIAL                                                                                      0072
  0054        ''THE FOLLOWING IS A BINARY SEARCH ALGORITHM WHICH DETERMINES IF A                                0073
  0054        PARTICULAR STEM IS IN THE DICTIONARY LIST OF CONTENT STEMS.''                                     0074
  0055                LPOINT=0 $                                                                                0075
  0056                INDEX=DICT+1 $                                                                            0076
  0057        SRCH10.  INDEX=(INDEX+1)/2 $                                                                      0077
  0058                IF (LPOINT+INDEX) GR RANGE $                                                              0078
  0059                GOTO SRCH10 $                                                                             0079
  0060              DIRECT                                                                                      0080
  0061              LDA      LPOINT                                                                             0081
  0062              ADA      INDEX                                                                              0082
  0063              SBA      1,DL                                                                               0083
  0064              ALS      1                                                                                  0084
  0065              LDAQ     CONSTM+1,AL                                                                        0085
  0066              CMPAQ    STORE+1                                                                            0086
  0067              TZE      SRTO                                                                               0087
  0068              TRC      SRCH30                                                                             0088
  0069              LDA      INDEX                                                                              0089
  0070              ASA      LPOINT                                                                             0090
  0071        SRCH30 LDA     1,DL                                                                               0091
  0072              SBA      INDEX                                                                              0092
  0073              TMI      SRCH10                                                                             0093
  0074              TRA      STEP1                                                                              0094
  0075              JOVIAL                                                                                      0095
  0075        ''THE FOLLOWING IS A MODIFIED D. L. SHELL SINKING SORT WHICH                                      0096
  0075        REORDERS THE D/W PAIRS OF A MASTER CONCORDANCE BY ASCENDING                                       0097
  0075        DOCUMENT NUMBER. THIS IS DONE ONLY AFTER  THE STEM HAS BEEN FOUND                                 0098
  0075        TO BE A CONTENT STEM.''                                                                           0099
```

Figure V. 20a. (Continued)

```
21400 01  04-23-71   08,664   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0076      SRT0.    LL=2 $                                                   710                0100
  0077      SRT1.    IF LL GQ PRCNT $                                         720                0101
  0078               GOTO SRT2 $                                              730                0102
  0079               LL=LL*2 $                                                740                0103
  0080               GOTO SRT1 $                                              750                0104
  0081      SRT2.    DD=(LL/2)-1 $                                            760                0105
  0084      SRT3.    II=0 $  JJ=0 $  KK=0 $                                   770                0106
  0085               DIRECT                                                   780                0107
  0086      SRT4   LDA      II                                                790                0108
  0087             EAX7     PAIRS+4,AL                                        800                0109
  0088             LDQ      0,7                                               810                0110
  0089             STQ      PAIR2                                             820                0111
  0090             ADA      DD                                                830                0112
  0091             EAX6     PAIRS+4,AL                                        840                0113
  0092             LDQ      0,6                                               850                0114
  0093             STQ      PAIR1                                             860                0115
  0094             LDA      PAIR1                                             870                0116
  0095             CMPA     PAIR2                                             880                0117
  0096             TMI      SRT7                                              890                0118
  0097               JOVIAL                                                   900                0119
  0098      SRT5.    IF II+DD+1 LS PRCNT $                                                       0120
  0099                BEGIN                                                   920                0121
  0100                 II=II+1 $                                                                 0122
  0101                 GOTO SRT4 $                                            940                0123
  0102                END                                                     950                0124
  0103               DD=(DD-1)/2 $                                            960                0125
  0104               IF DD LQ 0 $                                             970                0126
  0105               GOTO OUTPT $                                             975                0127
  0106               GOTO SRT3 $                                              980                0128
  0107               DIRECT                                                   990                0129
  0108      SRT7   LDA      0,7                                              1000                0130
  0109             STA      DUMMY1                                           1002                0131
  0110             LDA      0,6                                              1004                0132
  0111             STA      0,7                                              1006                0133
  0112             LDA      DUMMY1                                           1008                0134
  0113             STA      0,6                                              1010                0135
  0114               JOVIAL                                                  1012                0136
  0115               JJ=II-DD $                                              1014                0137
  0116               IF JJ LS 0 $                                            1016                0138
  0117               GOTO SRT5 $                                             1018                0139
  0118               KK=II $                                                 1020                0140
  0119               DIRECT                                                  1022                0141
  0120      SRT8   LDA      JJ                                               1024                0142
  0121             EAX7     PAIRS+4,AL                                       1026                0143
  0122             LDQ      0,7                                              1028                0144
  0123             STQ      PAIR2                                            1030                0145
  0124             LDA      KK                                               1032                0146
  0125             EAX6     PAIRS+4,AL                                       1034                0147
  0126             LDQ      0,6                                              1035                0148
  0127             STQ      PAIR1                                            1036                0149
```

Figure V. 20a. (Continued)

```
21400 01  04-23-71   08.664   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0128            LDA     PAIR1                                                   1038          0150
  0129            CMPA    PAIR2                                                   1040          0151
  0130            TZE     SRT5                                                    1042          0152
  0131            TPL     SRT5                                                    1044          0153
  0132            LDA     0,7                                                     1046          0154
  0133            STA     DUMMY1                                                  1048          0155
  0134            LDA     0,6                                                     1050          0156
  0135            STA     0,7                                                     1052          0157
  0136            LDA     DUMMY1                                                  1054          0158
  0137            STA     0,6                                                     1056          0159
  0138              JOVIAL                                                        1058          0160
  0139              JJ=JJ-DD $                                                    1060          0161
  0140              IF JJ LS 0 $                                                  1062          0162
  0141              GOTO SRT5 $                                                   1064          0163
  0142              KK=JJ+DD $                                                    1066          0164
  0143              GOTO SRT3 $                                                   1068          0165
  0143       ''THE FOLLOWING CODE WRITES THE MASTER CONCORDANCE OF A STEM                       0166
  0143       WHICH HAS BEEN FOUND IN THE DICTIONARY OF CONTENT STEMS                            0167
  0143       AND WHICH HAS HAD THE D/W PAIRS REORDER BY ASCENDING DOCUMENT                      0168
  0143       NUMBER ONTO THE OUTPUT TAPE, IF THE NUMBER OF CONCORDANCES OUTPUT IS               0169
  0143       EQUAL TO THE NUMBER OF DICTIONARY STEMS, EXECUTION IS TERMINATED.''                0170
  0144       OUTPT. IO003(4,REDUCE,0,PAIRS,PRCNT+3) $                                           0171
  0145            CNTOUT=CNTOUT+1 $                                                             0172
  0146            IF CNTOUT LS RANGE $                                                          0173
  0147            GOTO STEP1 $                                                    1110          0174
  0148       IEOF.  IN(2,STVC) $                                                  1200          0175
  0149            IN(2,RESORT) $                                                  1210          0176
  0150            OUT(2,REDUCE) $                                                 1220          0177
  0151            STOP $                                                          1230          0178
  0152            TERM GO $                                                       1240          0179
```

Figure V.20a. (Concluded)

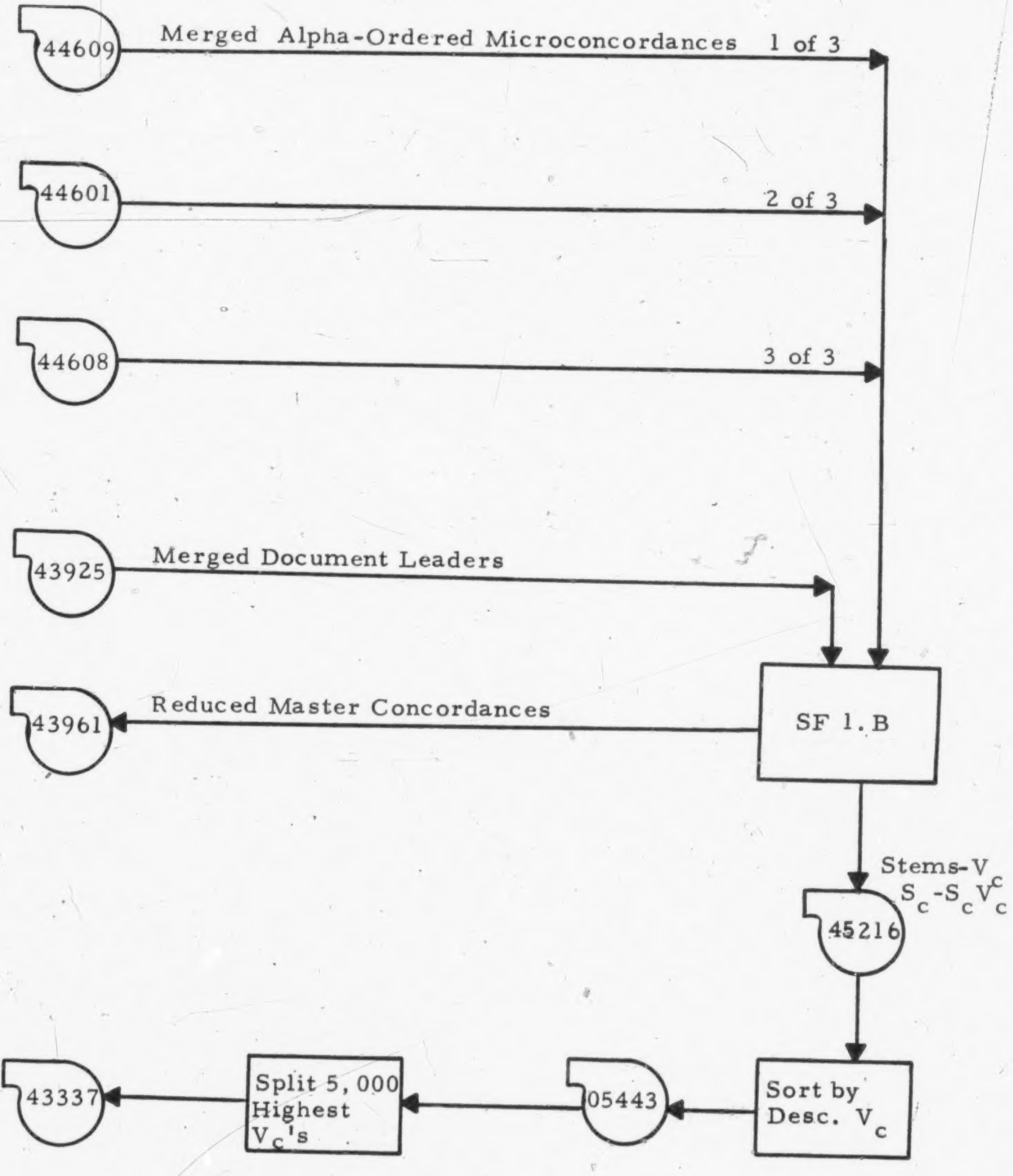

44609
Merged Alpha-Ordered Microconcordances
1 of 3
44601
2 of 3
44608
3 of 3
43925
Merged Document Leaders
43961
Reduced Master Concordances
SF 1.B
Stems-V_c
S_c-$S_cV_c^c$
45216
Sort by Desc. V_c
05443
Split 5,000 Highest V_c's
43337

Figure V.21. Statistical Filter Process Chart

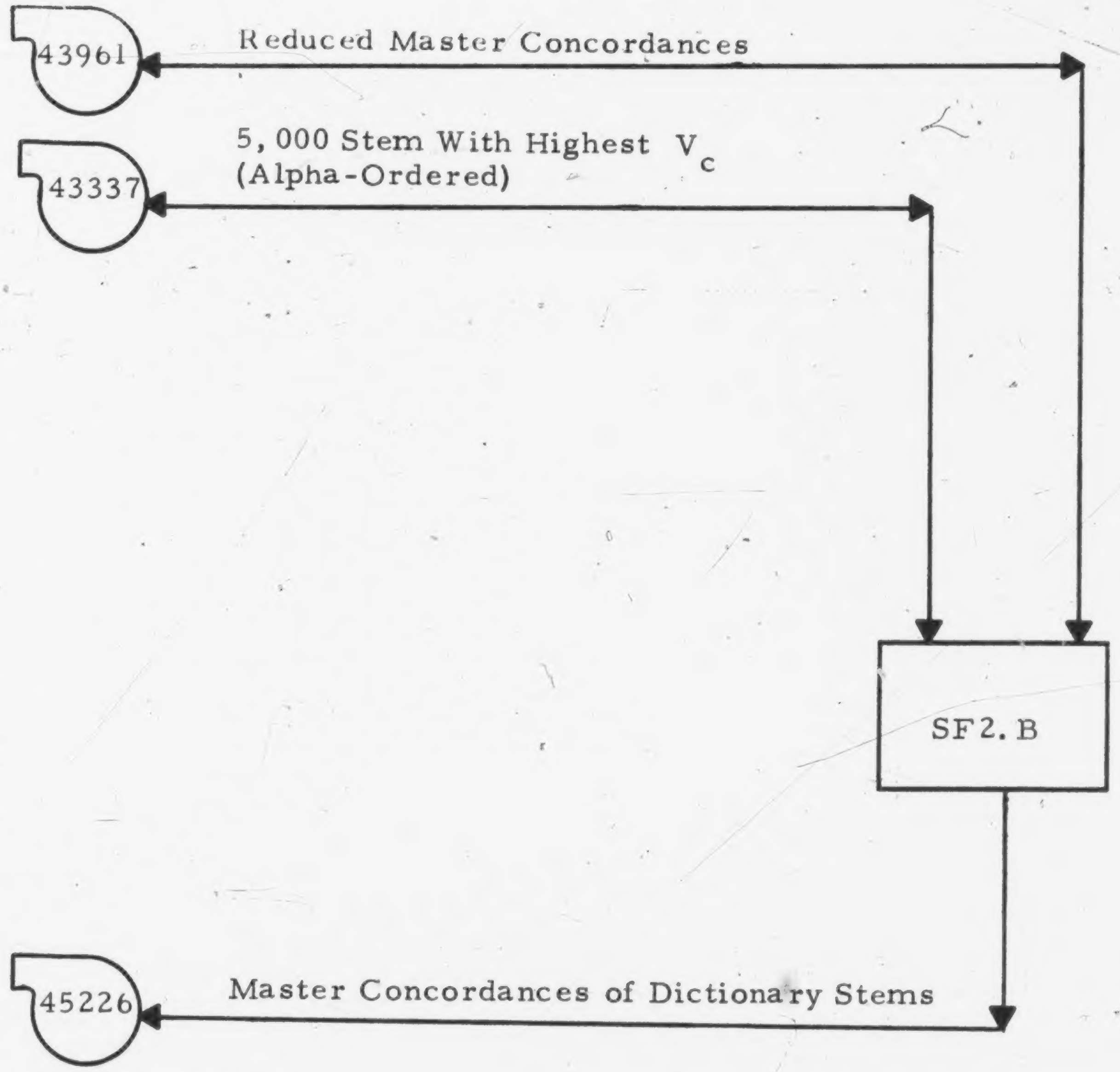

43961
Reduced Master Concordances
43337
5,000 Stem With Highest V_c (Alpha-Ordered)
SF2.B
45226
Master Concordances of Dictionary Stems

Figure V.21. (Concluded)

V.6 CONCEPT-DICTIONARY GENERATION (STEM CLUSTERING)

This sub-section describes the purposes and techniques of stem-clustering as applied to the data produced by the statistical filter and the micro-concordance generator.

V.6.1 Purposes

Concept dictionary generation is the next stage in automatic document characterization. In terms of the 5,000 content stems developed during statistical filtration, concept groups are generated during this pass. The method is as follows:

1) All the non-zero cosines between the 5,000 dimension-reduced stem concordances are computed, and are sorted in descending order by a system sort.

2) Next 1,500 concept centers are identified. By definition, a concept center is a stem concordance which correlates weakly with other concept center stems but strongly with non-concept-center stems. Some interesting experimental observations have been made on this process.

3) Lastly, each content-stem in the collection is correlated against the list of 1,500 concept-center concordances to obtain the mapping (dictionary entry) for each stem. Of course, those stems which are concept centers will have a correlation measure of 1.000 to themselves.

The product of this stem-clustering phase is the stem-dictionary with an entry for each stem. In turn, each stem is collected with the four highest concept-weight pairs produced during final correlation, to produce the random-access dictionary used during on-line retrieval. Because of the nature of the cosine computation, a certain number of content stems correlated only with themselves. For these cases a dummy concept (1501) has been used.

V.6.2 Method

As outlined above, the stem clustering algorithm consists of three separate functions:

PHASE I -	Routine CLUSTR.0, generates the dictionary stem scalar vector products (i.e., the cosine, in hyperspace, between two stems concordance).
PHASE II -	Routine CLUSTR.1, generates the distinct stems (i.e., the concept centers).
PHASE III -	Routine CLUSTR.2 builds the stem-concept/weight records which are used to construct the document and query concept vectors.

Input to CLUSTR.0 is the tape of dimension-reduced dictionary stem master concordances. The scalar product (i.e., $\cos\phi$) is calculated between every dictionary stem and stored on tape, $(\cos \Phi > 0.0)$. Each record of this tape contains the stem numbers and their correlation.

This tape is sorted by means of a system sort routine in order of descending $\cos \Phi$, the format required by routine CLUSTR.1, PHASE II of the stem clustering algorithm, in order to generate the Distinct Stems or Concept Numbers. Briefly, the distinct stems are stems which correlate strongly with other dictionary stems but correlate only weakly within themselves. The distinct stems are the concept centers used to characterize each dictionary stem by its related concept in the final phase of the clusterer, routine CLUSTR.2.

Routine CLUSTR.2 calculates the correlation between each dictionary stem and all concept centers and stores on tape the concept/weight pairs of the four concepts which correlate most strongly with that dictionary stem, along with the dictionary stem itself. This tape serves as input to the document concept vector generation routine.

V.6.2.1 Flowchart Comments - Routine CLUSTR.0. (refers to Figure V.22)

1) Open all input and output files.

2) Transfer to subroutine LOADAT, the routine which operates the array which contains the Dictionary Stem Master Concordances, and calculates the sum of the squares and the square root of the sum of the squares of the magnitude of the components of each concordance.

3) Transfer to subroutine, GENTAB, the code which calculates the correlation between all dictionary stems.

4) Close all input and output files. Terminate the job.

5) Read a master concordance from input file LOADER into the array CORE as specified by stem INDEX.

6) Test for an EOF on the input file LOADER. If an EOF is reached and the number of master concordances is less than specified by stem MAXLEN.

Premature EOF LOADER?

Yes ⇒ 7 No ⇒ 8

7) Issue a warning which states that an insufficient number of master concordances was found and terminate the job.

8) Calculate the sum of the squares and the square-root of the sum of the squares of the magnitude of the components of the current master concordance and store in array CORE with the master concordance.

9) Test for input of all master concordances

INDEX EQ MAXLEN?

Yes ⇒ 10; No ⇒ 5

10) Preset II to 0, JJ to 1, and CC to 0.0 II and JJ are counters used to reference master concordances in the array CORE and CC is the stem used to store the cosine correlation.

11) Calculate the dot product between two master concordances.

12) Test for a dot product of 0·0, indicating that the cosine correlation calculation need not be performed.

CC EQ 0·0 ?

Yes ⇒ 16; No ⇒ 13

13) Calculate the COSINE.

14) Construct the TRIPLET of II-JJ/CC.

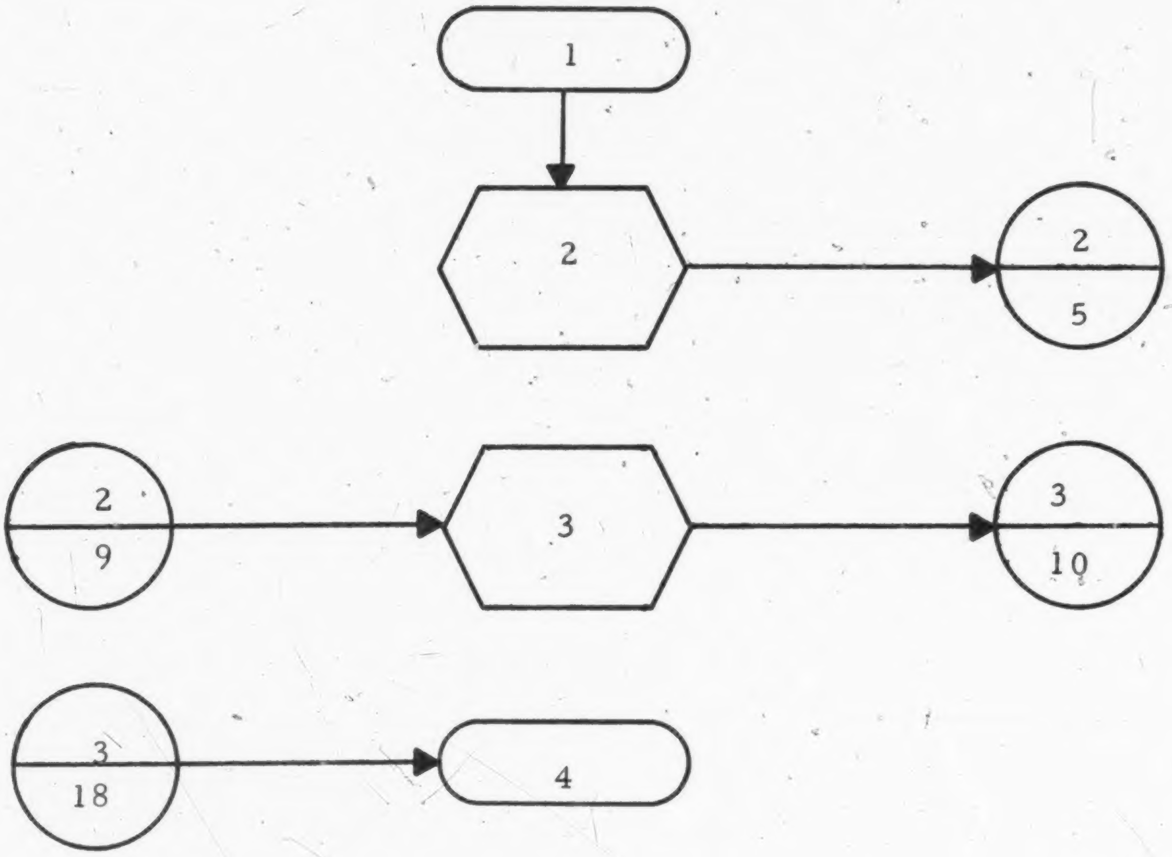

1
2
2
5
2
9
3
3
10
3
18
4

Figure V. 22. Flowchart CLUSTR. 0

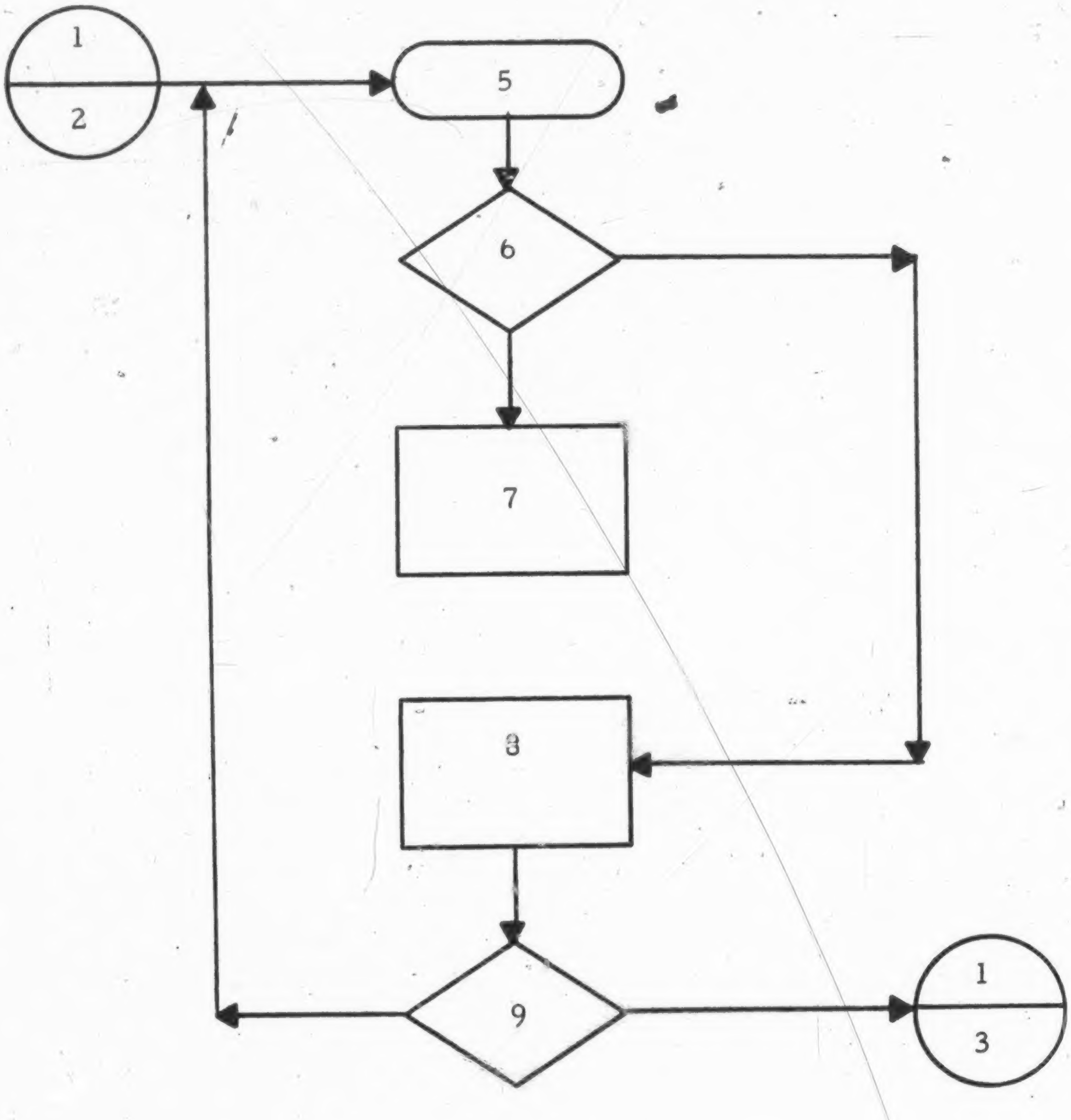


Figure V. 22. (Continued)

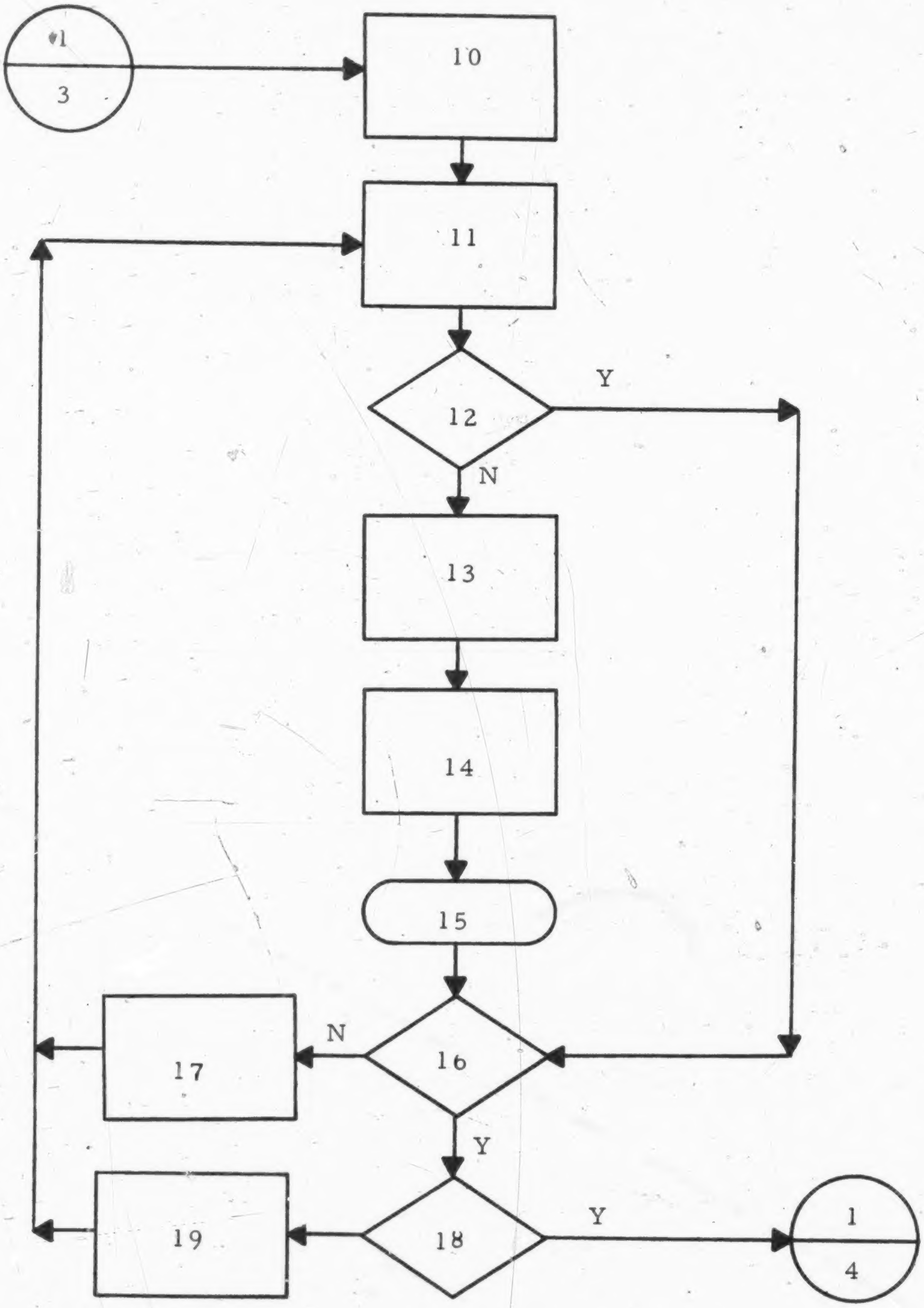

1
3
10
11
12
Y
N
13
14
15
16
N
17
Y
18
Y
19
1
4

Figure V. 22. (Continued)

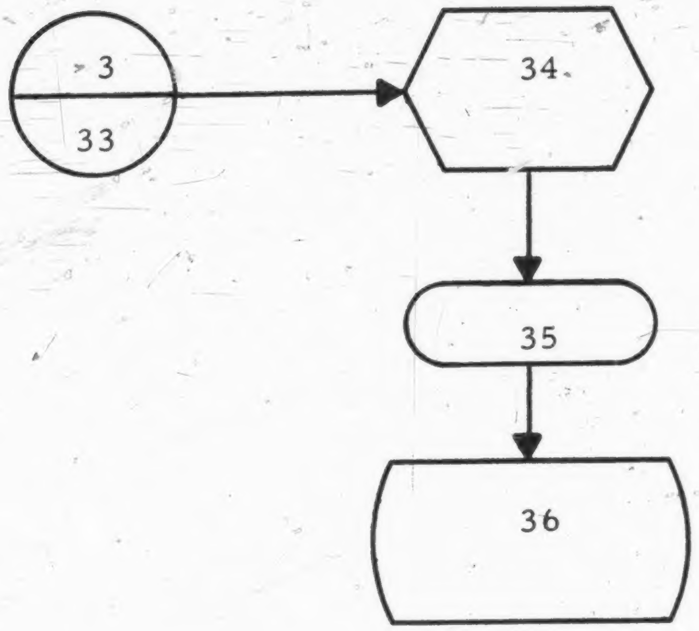


Figure V.22. (Concluded)

15) Write the TRIPLET on output file CORREL

16) Test for subscript JJ at the limit of its range.
JJ at limit?
Yes ⇒18 No ⇒17

17) JJ + 1 ⇒JJ.

18) Test for subscript II at the limit of its range.
II at limit?
Yes ⇒4 No ⇒19

19) II + 1 ⇒II
II + 1 ⇒JJ

V.6.2.2 Production Run Summary: CLUSTR.0.

1) 12,500,000 correlations calculated in 2.9922 hours CPU time ELAPSED TIME : 3 hours

2) 82,443 correlations > 0.0 were calculated by system sort routine

0.0477 hours CPU time required for system sort routine

V.6.2.3 Operational Procedure: Routine CLUSTR.0.

1) Preset DEFINE "MAXCOR" to the number of words of storage required to store the master concordances.

Preset DEFINE "MAXLEN" to the number of master concordances to be input.

2) Assign tapes:

I/O File	File Code	Description
LOADER	30	Input of master concordances
CORREL	33	Output of correlation triplets

V.6.2.4 Flowchart Comments - Routine CLUSTER. 1. Refers to Figure V.23

1) Zero-fill ARRAY FLAGS, the array which has assigned 1 bit per stem; if a bit is turned "on" (=1), the corresponding stem has been deleted from consideration as a distinct stem.

Set all stems in table DSTAB to -1; this table contains the number of each distinct stem.

2) Open all input and output files

3) Set stem PTL, the index to the next unused entry in the distinct stem table, to 0; set SS, the number of distinct stems to 0.

4) Set the two status flags, ISTATS and JSTATS, to condition V(OFF). If either is set to V(ON), this indicates that either stem I or J of the current CORRELATION TRIPLET has been found in the "deleted from consideration" array, FLAGS. Read a TRIPLET from input file LOADER.

5) Search for both stem I and J of the current TRIPLET in array FLAGS.

6) If both ISTATS and JSTATS are at condition V(ON), neither stem I nor stem J of the current TRIPLET can be Distinct Stems; therefore, return to fetch this next TRIPLET.

ISTATS EQ V(ON) AND JSTATS EQ V(ON)?

Yes ⇒ 4 No ⇒ 7

7) Set item RR to 0, this item is the term which is used to sequentially index the distinct stem table during a search for the presence of stem I or J of the current TRIPLET. Set IFLAG and JFLAG to V(TRUE); if either stem I or J is found in the distinct stem table, IFLAG or JFLAG is set to condition V(FALSE), respectively.

8) Check to see if all entries of the distinct stem table have been tested against the current triplet.

RR EQ PTL?

Yes ⇒ 17 No ⇒ 9

9) Stem I of the current TRIPLET equal to the entry of the distinct stem table indexed by RR?

Yes ⇒ 10 No ⇒ 11

10) V(FALSE) ⇒ IFLAG.

11) Stem J of the current triplet TRIPLET equal to the entry of the distinct stem table indexed by RR?

Yes ⇒ 12 No ⇒ 13

12) V(FALSE) ⇒ JFLAG

RR ⇒ RJ. (RJ is the pointer to the distinct stem table, if both stems of the current TRIPLET are distinct stems, the entry indexed by RJ is set to -1.)

13) Test for either stem having a status V(TRUE), indicating that it is still a distinct stem candidate.

IFLAG or JFLAG EQ V(TRUE)?

Yes ⇒14 No ⇒15

14) RR + 1 ⇒R.

15) Enter -1 in the correct entry of the distinct stem table and decrement the count of distinct stems.

16) Call CLOSE "FLAGIT" in order to set the bit in the FLAGS array as specified by item XXX.

17) IFLAG NQ JFLAG?

Yes ⇒ 18 No ⇒22

This is the test for either I or J (but not both) in the distinct stem table. If both are in the distinct stem table, IGLAG and JFLAG EQ V(FALSE), this code will not be reached.

18) Test for stem I in the distinct stem table and the bit in array FLAGS set to V(OFF) for stem J indicating that stem J has not been deleted from consideration as a distinct stem.

IFLAG EQ V(FALSE) AND JSTATS EQ V(OFF)?

Yes ⇒19 No ⇒20

19) Call FLAGIT to flag stem J.

20) Test for stem J in the distinct stem table and the bit in array FLAGS set to V(OFF) for stem I.

JFLAG EQ V(FALSE) AND ISTATS EQ V(OFF)?

Yes ⇒ 21 No ⇒ 4

21) Call FLAGIT to flag stem I.

22) Check if stem I has been deleted from consideration as a distinct stem.

ISTATS EQ V(ON)?

Yes ⇒29 No ⇒23

23) Enter stem I in the distinct stem table; increment SS, the counter of distinct stems; increment PTL, the pointer to the next vacant position in the distinct stem table.

24) Test stem PTL against stem RANGE, a preset value which specifies the maximum size of the distinct stem table.

PTL EQ RANGE?

Yes ⇒25 No ⇒27

25) Call CLOSE "COMPAC" which packs the distinct stem table.

26) Reset the PTL parameter to SS, the number of distinct stems, upon exiting the COMPAC routine; SS is reset in COMPAC during the compacting procedure.

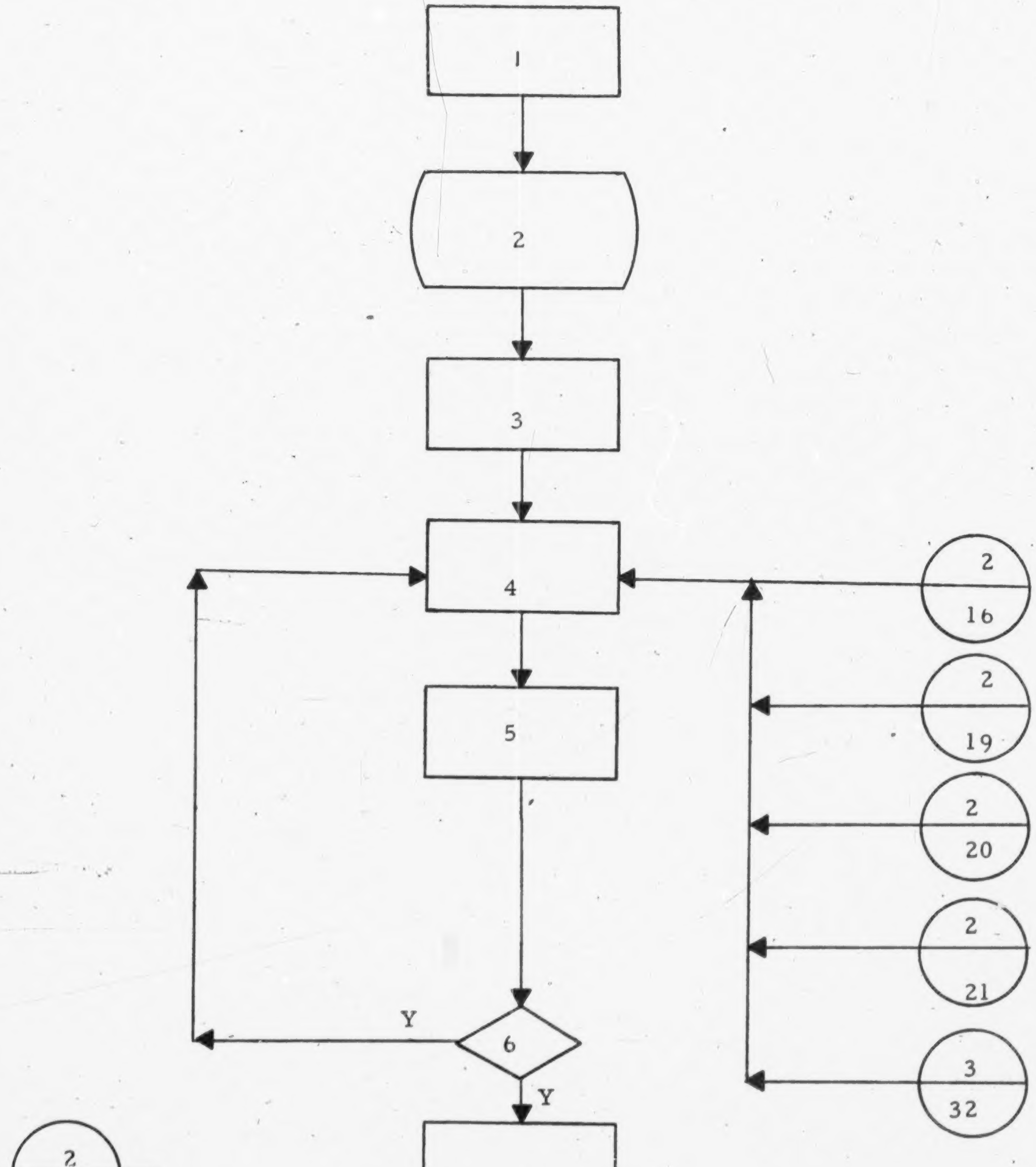

1
2
3
4
5
6
Y
Y
7
8
Y
N
2
16
2
19
2
20
2
21
3
32
2
14
2
17
2
9

Figure V. 23. Flowchart CLUSTR. 1

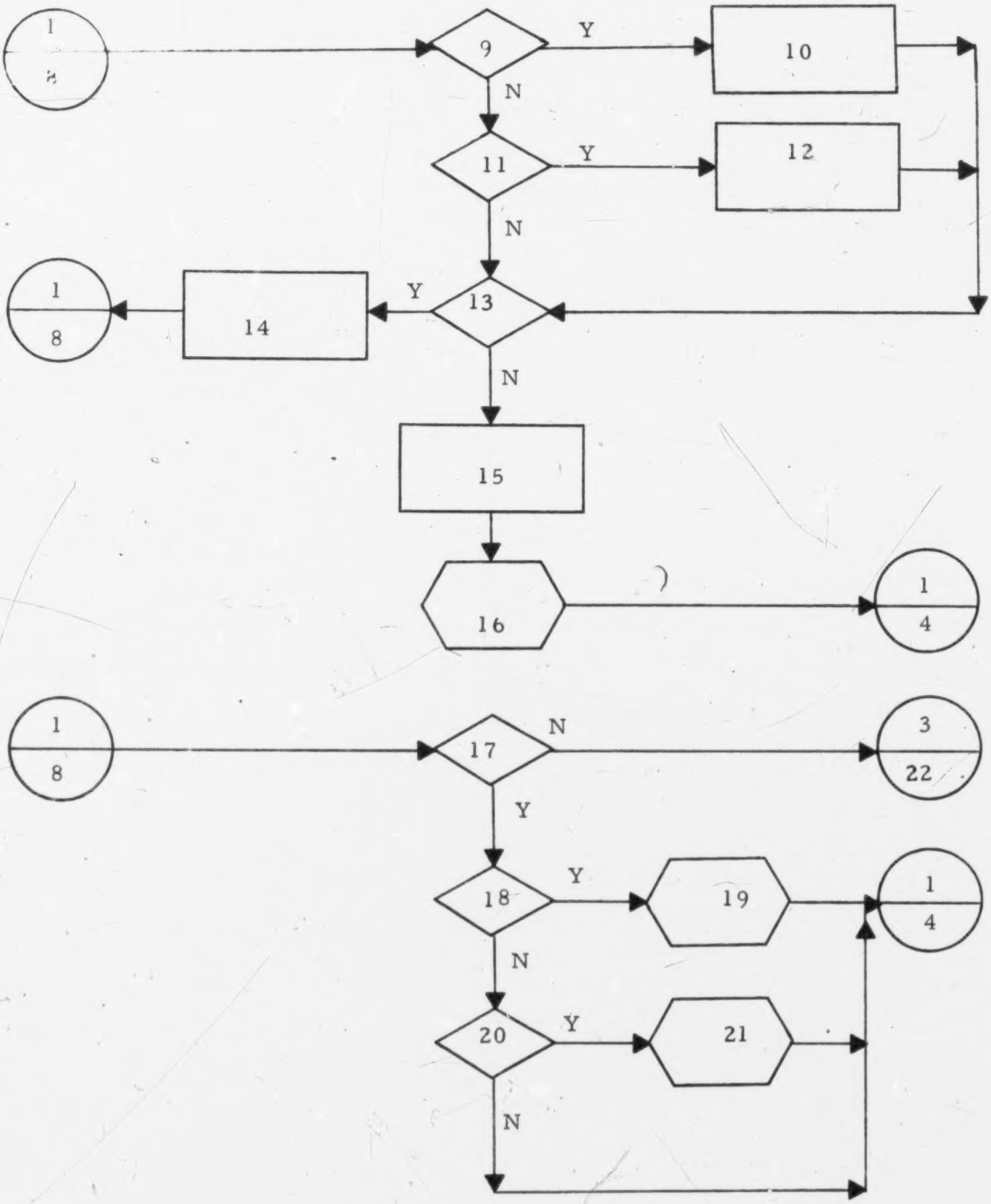

1
8
9
Y
10
N
11
Y
12
N
1
8
14
Y
13
N
15
16
1
4
1
8
17
N
3
22
Y
18
Y
19
1
4
N
20
Y
21
N

Figure V. 23. (Continued)

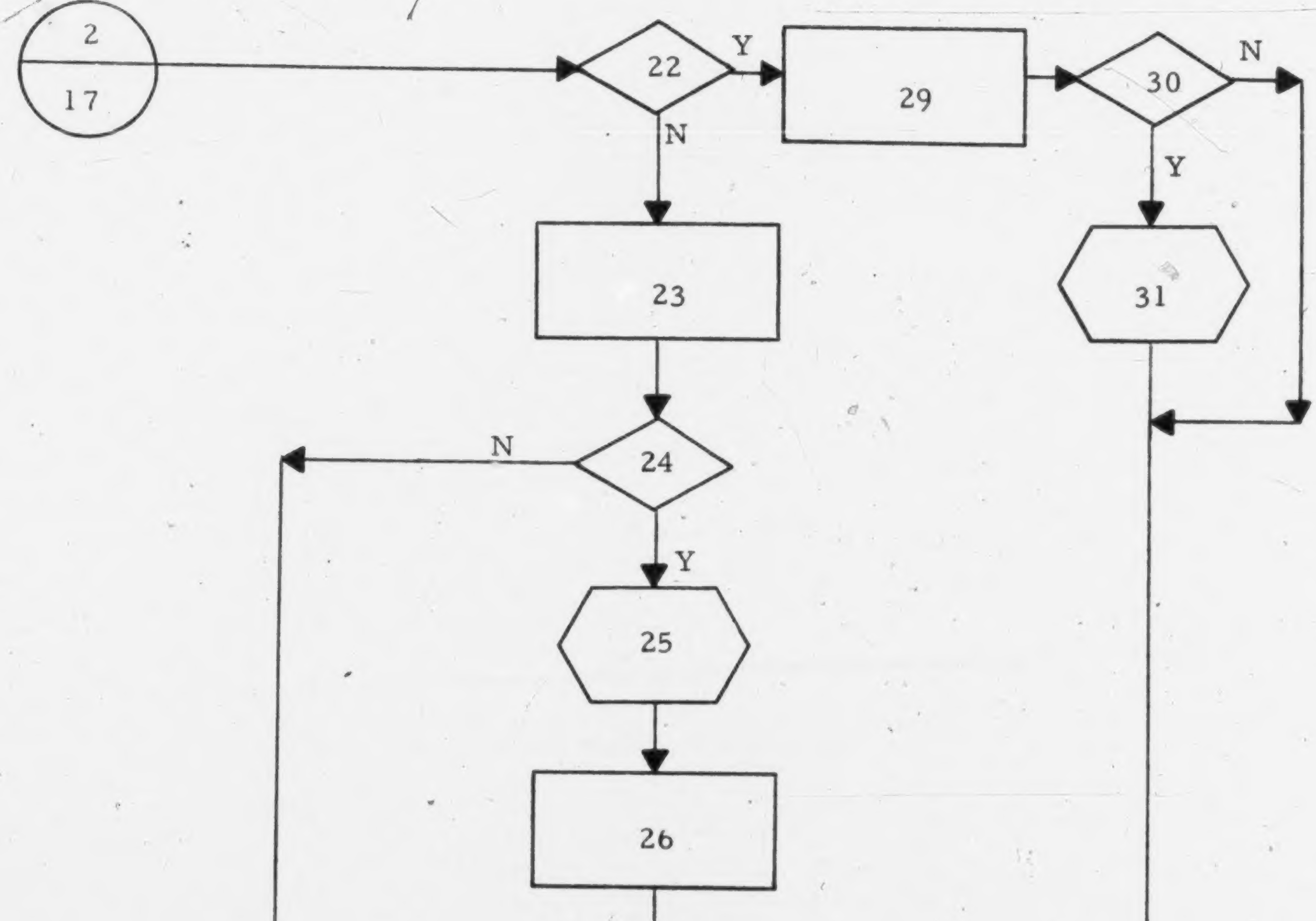

2
17
22
Y
N
29
30
N
Y
23
31
24
N
Y
25
26
27
28
N
1
4
Y
32
33
4
34

Figure V. 23. (Concluded)

27) Test for JSTATS set to V(OFF) before flagging J in the "deleted from consideration" array FLAGS.

JSTATS EQ V(OFF)?

Yes ⇒ 28 No ⇒ 32

28) Flag stem J in array FLAGS.

29) Enter stem J in the distinct stem table; increment SS; increment PTL.

30) Test stem PTL against stem RANGE, a preset value which specifies the maximum size of the distinct stem table.

PTL EQ RANGE?

Yes ⇒31 No ⇒32

31) Compare SS, the number of distinct stems, against NODIST, a preset value which designates the number of distinct stems sought.

SS LS NODIST?

Yes ⇒4 No ⇒33

32) Call COMPAC to compact the distinct stem table.

33) Call CLOSE "SORTDS" which employes a D. L. SHELL SINKING SORT to order the distinct stem table in order of ascending stem number.

34) Write the distinct stem numbers on output file DSTABL.

35) Close all I/O files and terminate the job.

V.6.2.5 Operational Procedure - Routine CLUSTR.1.

1) Preset DEFINE: "NUMFLG", the number of words required by array FLAGS in order to assign one bit per dictionary stem.

"RANGE", the size, in words, of the distinct stem table.

"NODIST", the number of distinct stems desired.

2) Assign Tapes:

I/O File	File Code	Description
LOADER	30	Input of CORRELATION TRIPLETS sorted by descending cos Φ.
DISTABL	32	Output of Distinct Stems ordered by descending stem number.

V.6.2.6 Production Run Summary - Routine CLUSTR.1.

CPU TIME : 0.0600 hours

ELAPSED TIME : 0.180 hours.

V.6.2.7 Flowchart Comments - Routine CLUSTR.2. Refers to Figure V.24

1) Open all I/O files.

2) Read the number of distinct stem numbers specified by item NODIST from input file CENTER and store in array DISTCT.

3) Build the array of distinct stem master concordances, CONCRD, by reading the entire 5,000 dictionary concordances from input file LOADER and storing only those concordances whose stem number is found in DISTCT. During this loading process, calculate the scan of the square and the square-root of the sum of the squares of the magnitude of the components of each master concordance and store these two values with the concordance in array CONCRD.

4) 0 ⇒INDEX

5) Rewind input file LOADER.

6) Read a dictionary master concordance from input file LOADER and store in array INBUF.

7) Call FUNCTION "MAG" which calculates the sum of the squares of the magnitude of the components of the current dictionary concordance. This value is returned in stem TMPMAG.

8) Store TMPMAG and $\sqrt{\text{TMPMAG}}$ in the array with the current master concordance.

9) Calculate the correlation between the current master concordance and all concept stem (distinct stem) master concordances. If the correlation is greater than 0.0, enter the concept center number and the correlation value in table CORTAB and index by counter INDEX.

10) Call Procedure SORTCT, a modified D. L. Shell Sinking Sort which orders the Concept/Weight pairs in table CORTAB by descending WEIGHT.

11) Construct the cluster which consists of the hollerith stem and the four Concept/Weight pairs which correlate most strongly with the current dictionary stem master concordance. This record is stored in array VECTOR.

12) Write the cluster on output file CONCPT.

13) If all dictionary stem master concordances have not been processed, transfer to fetch next concordance.

Processing completed?

Yes ⇒ 14 No ⇒ 6

14) Close all I/O files and terminate job.

V.6.2.8 Operational Procedure: Routine - CLUSTR.2.

1) Preset DEFINES:

NODIST - the number of distinct stems

MAXLEN - the number of dictionary stem master concordances to be input.

MAXCOR - the number of words of storage needed for the concept center master concordances.

CLUSIZ - the number of CONCEPT/WEIGHT pairs per cluster.

VECSIZ - the number of words assigned to the output cluster (normally: VECSIZ = CLUSIZ + 2).

2) Assign tapes:

I/O File	File Code	Description
CENTER	10	Input of Distinct Stem numbers
LOADER	11	Input of Master Concordances
CONCPT	12	Output of Dictionary Stems and CONCEPT/WEIGHT pairs

V.6.2.9 Production Run Summary - Routine CLUSTR.2.

CPU time: 1.8321 hours

Elapsed time: ≈ 3 hours

Number of Dictionary: 5000

Number of Distinct Stems: 1500

≈1,700 dictionary stems correlated with less than 4 concepts of these, about 10% had zero correlations.

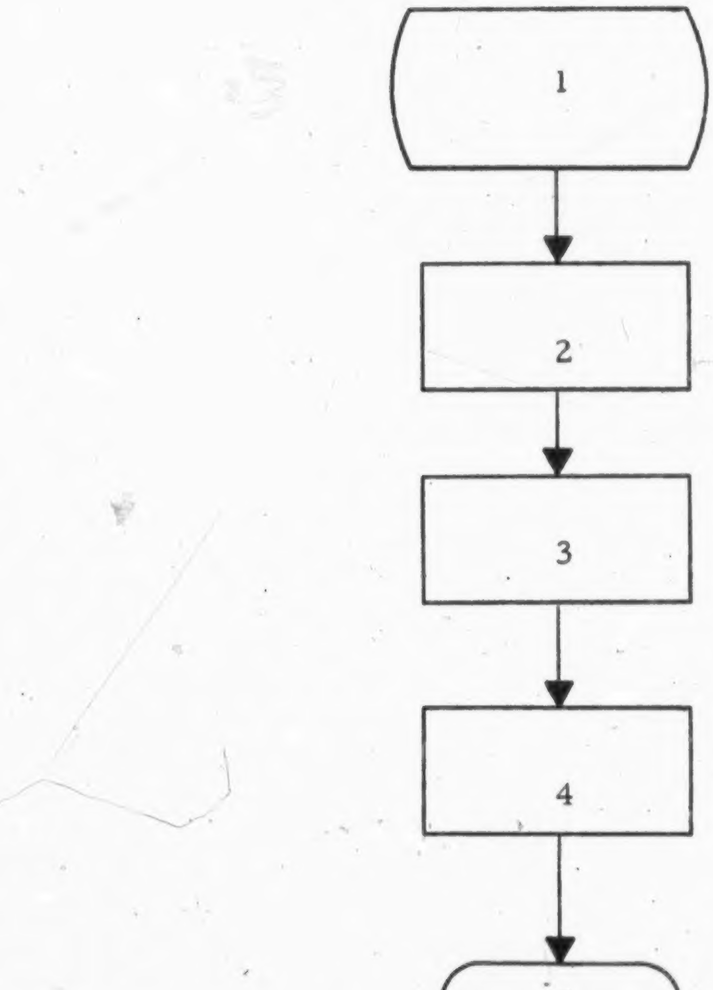

1
2
3
4
5
2
13
6
7
8
2
9

Figure V.24. Flowchart CLUSTR.2

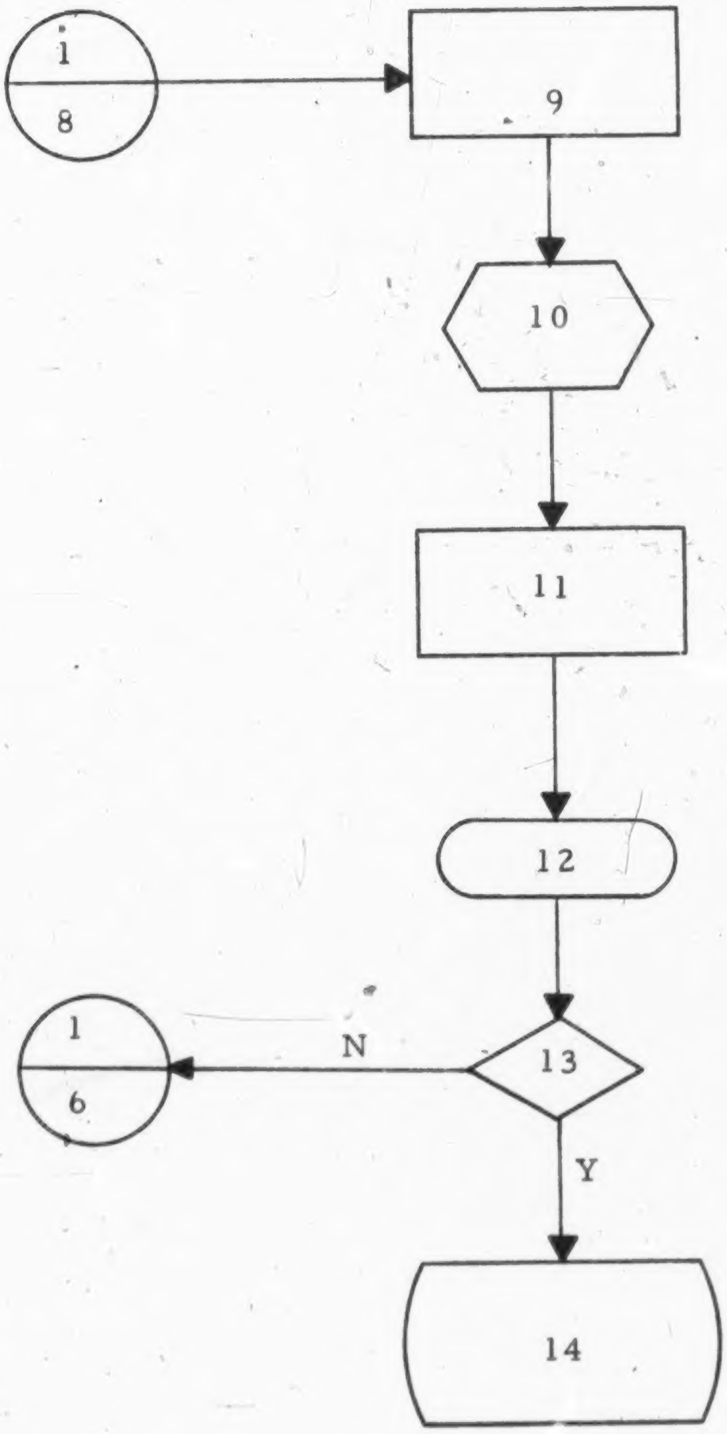

1
8
9
10
11
12
N
13
1
6
Y
14

Figure V. 24. (Concluded)

```
12733 01  04-29-71   18.101   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM CLUSTR.0
  0000       START                                                              70
  0000       ''                                                                 80
  0000       ***************************************************************   90
  0000                      D E F I N I T I O N S                             100
  0000       ***************************************************************  110
  0000       ''
  0001             DEFINE MAXCOR ''175000'' $
  0002             DEFINE MAXLEN ''5000'' $                                     192
  0003             DEFINE INTEG ''I 36 S'' $                                    194
  0004             DEFINE NOR ''I 35 U'' $                                      380
  0004       ''                                                                390
  0004       ***************************************************************  490
  0004                   ARRAYS,TABLES, AND FILE DESCRIPTIONS                  410
  0004       ***************************************************************  420
  0004       ''
  0005             ARRAY CORE 35 1 INTEG $
  0006             ARRAY CORF 35 1     F $
  0007             OVERLAY CORE=CORF $
  0008             TABLE ALOCOR R MAXCOR S 1 $
  0008                   BEGIN
  0009                   ITEM ALOC1 H 6 0 0 $
  0010                   END
  0010             FILE CORREL ''OUTPUT OF CORRELATIONS''                       711
  0010                   V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R33 $      720
  0011             FILE LOADER ''INITIAL CONCORDANCES OR RESTART DATA.''        730
  0011                   V(NORM) V(NULL) V(SEG)  V(EOF)  V(LERR) V(PERR) R30 $  791
  0012             FILE PRNT  ''OUTPUT TO PRINTER''                             792
  0012                   V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R06 $     3490
  0013       PROCESS.                                                          3665
  0013             OUT(1,PRNT) $
  0014             OUT(1,CORREL) $                                             3690
  0015             IN(1,LOADER) $                                              3700
  0016             LOADAT $
  0017             GENTAB $
  0018             OUT(2,PRNT) $
  0019             OUT(2,CORREL) $
  0020             IN(2,LOADER) $
  0021             STOP $                                                     10011
  0021       ''************************************************************''  10012
  0021       ''                         LOADAT                               ''  10020
  0021       ''************************************************************''  10025
  0022       PROC LOADAT $                                                 00010030
  0023             ITEM INDEX I 36 S $                                     00010040
  0024             ITEM TMPMAG F $                                              280
  0025             ITEM WARN1 H 36 P 36H(CONCORDANCES INPUT LESS THAN MAXLEN ) $ 00010050
  0026       BEGIN                                                         00010060
  0027             INDEX=0 $                                                  10070
  0028       LD1.  IO20(0,LOADER,J,CORE($0,INDEX$),33) $                      10080
  0029             IF LOADER EQ V(EOF) AND INDEX LS MAXLEN $
```

Figure V.25. CLUSTR.0 Listing

```
12733 01  04-29-71   18.101    JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

   0030                     BEGIN                                                  10091      0050
   0031                     IC1012(3,PRNT,0,WARN1,6) $                             10092      0051
   0032                     STOP $                                                 10084      0052
   0033                     END                                                    10085      0053
   0034             TMPMAG = MAG(CORE($0,INDEX$)) $                                10090      0054
   0035             CORE($34,INDEX$) = TMPMAG $                                    10100      0055
   0036             CORE($33,INDEX$) = TMPMAG**.5 $                                10110      0056
   0037             INDEX=INDEX+1 $                                             00010120      0057
   0038             IF INDEX LS MAXLEN $                                           10125      0058
   0039             GOTO L01 $                                                     10130      0059
   0040      END                                                                00010160      0060
   0040      ''***************************************************************''    2170      0061
   0040      ''                          GENTAB                               ''    2171      0062
   0040      ''***************************************************************''    2172      0063
   0040      ''PROCEDURE GENTAB GENERATES THE CORRELATION TABLE.''                  2180      0064
   0041      PROC GENTAB $                                                                    0065
   0042             ITEM    II INTEG$                                               2200      0066
   0043             ITEM    JJ INTEG $                                              2210      0067
   0044             ITEM    CC " $                                                  2230      0068
   0045             ITEM WT1 NOR $                                                            0069
   0046             ITEM WT2 NOR $                                                            0070
   0047             ITEM DOC1 NOR $                                                           0071
   0048             ITEM DOC2 NOR $                                                           0072
   0049             ITEM DOTPRD NOR $                                                         0073
   0050             ITEM ICOUNT NOR $                                                         0074
   0051             ITEM JCOUNT NOR $                                                         0075
   0052             ITEM TRIPLT H 12 $                                                        0076
   0053      BEGIN                                                                  2430      0077
   0054             II = 0 $                                                        2450      0078
   0055             JJ = 1 $                                                        2460      0079
   0056             CC=0.0 $                                                        2480      0080
   0057      GEN11.                                                                           0081
   0057             DIRECT                                                                    0082
   0058             LDQ     II                                                                0083
   0059             MPY     35,DL                                                             0084
   0060             EAX4    CORE,QL                                                           0085
   0061             EAX5    CORE+3,QL                                                         0086
   0062             LDQ     JJ                                                                0087
   0063             MPY     35,DL                                                             0088
   0064             EAX6    CORE,QL                                                           0089
   0065             EAX7    CORE+3,QL                                                         0090
   0066             STZ     ICOUNT                                                            0091
   0067             STZ     JCOUNT                                                            0092
   0068             STZ     DOTPRD                                                            0093
   0069             STZ     TRIPLT                                                            0094
   0070             STZ     TRIPLT+1                                                          0095
   0071      DOT0   LDA     ICOUNT                                                            0096
   0072             CMPA    0,4                                                               0097
   0073             TZE     DOT4                                                              0098
   0074             LDA     JCOUNT                                                            0099
```

Figure V.25. (Continued)

```
12733 01  04-29-71   18.101   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0075                CMPA    0,6                                                              0100
  0076                TZE     DOT4                                                             0101
  0077                LDA     0,5                                                              0102
  0078                STCA    WT1,07                                                           0103
  0079                ARL     18                                                               0104
  0080                STA     DOC1                                                             0105
  0081                LDA     0,7                                                              0106
  0082                STCA    WT2,07                                                           0107
  0083                ARL     18                                                               0108
  0084                STA     DOC2                                                             0109
  0085         DOT3   LDA     DOC1                                                             0110
  0086                CMPA    DOC2                                                             0111
  0087                TNZ     DOT1                                                             0112
  0088                LDQ     WT1                                                              0113
  0089                MPY     WT2                                                              0114
  0090                ASQ     DOTPRD                                                           0115
  0091                ADX5    1,DU                                                             0116
  0092                ADX7    1,DU                                                             0117
  0093                AOS     ICOUNT                                                           0118
  0094                AOS     JCOUNT                                                           0119
  0095                TRA     DOT0                                                             0120
  0096         DOT1   TPL     DOT2                                                             0121
  0097                AOS     ICOUNT                                                           0122
  0098                LDA     ICOUNT                                                           0123
  0099                CMPA    0,4                                                              0124
  0100                TZE     DOT4                                                             0125
  0101                ADX5    1,DU                                                             0126
  0102                LDA     0,5                                                              0127
  0103                STCA    WT1,07                                                           0128
  0104                ARL     18                                                               0129
  0105                STA     DOC1                                                             0130
  0106                TRA     DOT3                                                             0131
  0107         DOT2   AOS     JCOUNT                                                           0132
  0108                LDA     JCOUNT                                                           0133
  0109                CMPA    0,6                                                              0134
  0110                TZE     DOT4                                                             0135
  0111                ADX7    1,DU                                                             0136
  0112                LDA     0,7                                                              0137
  0113                STCA    WT2,07                                                           0138
  0114                ARL     18                                                               0139
  0115                STA     DOC2                                                             0140
  0116                TRA     DOT3                                                             0141
  0117                JOVIAL                                                                   0142
  0118         DOT4.                                                                           0143
  0118                CC=DOTPRD $                                                              0144
  0119                IF CC   EQ 0.0 $                                                         0145
  0120                GOTO CENT3 $                                                             0146
  0121                CC=CC/(CORF($33,II$)*CORF($33,JJ$)) $                                    0147
  0122                DIRECT                                                                   0148
  0123                LDA     II                                                               0149
```

Figure V. 25. (Continued)

```
12733 01  04-29-71   18.101   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0124           ALS     18                                                     0150
 0125           ADA     JJ                                                     0151
 0126           STA     TRIPLT                                                 0152
 0127           LDA     CC                                                     0153
 0128           STA     TRIPLT+1                                               0154
 0129           JOVIAL                                                         0155
 0130           IO100(4,CORREL,0,TRIPLT,2) $                                   0156
 0131       GENT3.                                                     2930    0157
 0131           IF JJ EQ MAXLEN-1 $                                    3260    0158
 0132           GOTO GENT7 $                                           3270    0159
 0133           DIRECT                                                 3280    0160
 0134           AOS     JJ                                             3290    0161
 0135           TRA     GENT1            GOTO GENT1                    3300    0162
 0136           JOVIAL                                                 3310    0163
 0137       GENT7.                                                     3320    0164
 0137           IF II EQ MAXLEN-2 $                                    3330    0165
 0138           RETURN $                                                       0166
 0139           DIRECT                                                 3360    0167
 0140           LDA     II               II=II+1                       3370    0168
 0141           ADA     1,DL                                           3380    0169
 0142           STA     II                                             3390    0170
 0143           LDA     1,DL             JJ=II+1                       3400    0171
 0144           STA     JJ                                             3410    0172
 0145           TRA     GENT1            GOTO GENT1                    3420    0173
 0146           JOVIAL                                                 3430    0174
 0147       END ''OF CORRELATION-TABLE GENERATION''                    3470    0175
 0147       ''*********************************************************''  1691    0176
 0147       ''                          MAG                            ''  1692    0177
 0147       ''*********************************************************''  1693    0178
 0148       PROC    MAG(VEC) $                                         1700    0179
 0148       ''COMPUTES THE SUM OF THE SQUARES OF THE VECTOR COMPONENTS''  1710    0180
 0149           ARRAY   VEC 35 I 36 S $ BEGIN END                      1720    0181
 0150           ITEM    TEMPCL F $                                     1730    0182
 0151           ITEM    WTPRT I 36 S $                                 1740    0183
 0152           ITEM TEMP I 36 S  $                                    1750    0184
 0153           ITEM MAG F $                                           1760    0185
 0154           ITEM SEVENS I 36 S P O(000000777777) $                 1763    0186
 0155           BEGIN ''A''                                            1770    0187
 0156           TEMPCL=0.0 $                                           1780    0188
 0157           FOR I = 3,1,VEC($0$)+2 $                               1790    0189
 0158                   BEGIN                                          1800    0190
 0159                   TEMP=VEC($I$) $                                1810    0191
 0160                   DIRECT                                         1820    0192
 0161           LDA     TEMP                                           1830    0193
 0162           ANA     SEVENS                                         1840    0194
 0163           STA     WTPRT                                          1850    0195
 0164                   JOVIAL                                         1860    0196
 0165                   TEMPCL=TEMPCL+(WTPRT*WTPRT) $                  1870    0197
 0166                   END                                            1880    0198
 0167           MAG=TEMPCL $                                           1890    0199
```

Figure V.25. (Continued)

```
12733 01  04-29-71   18.101   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0168      END ''OF VECTOR MAGNITUDE COMPUTATION''                              1910              0200
 0169      TERM PROCESS $                                                      11750              0201
```

Figure V.25. (Concluded)

```
12735 01  04-29-71   18.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM CLUST1 ''  STEM CLUSTRRING  ''
  0000              START                                                      60
  0000        ..                                                               70
  0000        ***********************************************************      80
  0000                   D E F I N I T I O N S                                 90
  0000        ***********************************************************     100
  0000        ..                                                              110
  0001              DEFINE NUMFLG ''400'' $ ''(MAXLEN/36)+1=NENT FLAGS''
  0002              DEFINE RANGE ''2000'' $ ''SIZE OF DISTINCT STEM TABLE.''
  0003              DEFINE NODIST ''1500'' $ ''NO. OF DISTINCT STEMS.''
  0004              DEFINE INTEG ''I 36 S'' $                                 192
  0005              DEFINE NOR ''I 35 U'' $                                   194
  0005        ''*********************************************************''   220
  0005        ''           I T E M  D E S C R I P T I O N S              ''   230
  0005        ''*********************************************************''   240
  0006              ITEM PTL INTEG P 0 $                                      250
  0007              ITEM SS  INTEG P 0 $                                      251
  0008              ITEM RR INTEG P 0 $
  0009              ITEM XXX INTEG P 0 $                                      252
  0010              ITEM JJJ INTEG P 0 $
  0011              ITEM III INTEG P 0 $
  0012              ITEM TRIPLT H 12 $
  0013              ITEM CONTRM INTEG P 0 $                                   253
  0014              ITEM IFLAG S V(FALSE) V(TRUE) $                           330
  0015              ITEM    JFLAG S V(FALSE) V(TRUE) $                        340
  0016              ITEM ISTATS S V(OFF) V(ON) P V(OFF) $
  0017              ITEM JSTATS S V(OFF) V(ON) P V(OFF) $
  0018              DIRECT                                                    372
  0019        INSTO ALN    0                                                  373
  0020              JOVIAL                                                    374
  0020        ..                                                              380
  0020        ***********************************************************     390
  0020                   ARRAYS,TABLES, AND FILE DESCRIPTIONS                 400
  0020        ***********************************************************     410
  0020        ..                                                              420
  0021              ARRAY FLAGS NUMFLG INTEG $  BEGIN END                     430
  0022              TABLE DSTAB R RANGE $ ''DISTINCT STEM NUMBERS''           540
  0022                    BEGIN                                               550
  0023                    ITEM DS I 36 S $                                    560
  0024                    END                                                 570
  0024              FILE LOADER ''INPUT OF CORRELATION TRIPLETS.''
  0024                    V(NORM) V(NULL) V(SEG)  V(EOF)  V(LERR) V(PERR) R30 $    730
  0025              FILE DSTABL ''OUTPUT OF DISTINCT-STEM TABLE''
  0025                    V(NORM) V(NULL) V(SEG)  V(EOF)  V(LERR) V(PERR) R32 $    770
  0026              FILE PRNT  ''OUTPUT TO PRINTER''                          791
  0026                    V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R06 $   792
  0027        PROCESS.                                                       3490
  0027              FOR I = 0,1,RANGE-1 $                                    3570
  0028                    BEGIN                                              3580
  0029                    DS($I$)=-1 $                                       3605
```

Figure V.26. CLUSTR.1 Listing

```
12735 01  04-29-71   18.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0030                    END                                                      3610          0050
 0031              FOR I=0,1,NUMFLG-1 $                                           3652          0051
 0032                    BEGIN                                                    3654          0052
 0033                    FLAGS($I$)=0 $                                           3656          0053
 0034                    END                                                      3658          0054
 0035              OUT(1,PRNT) $                                                  3665          0055
 0036              OUT(1,DSTABL) $                                                3670          0056
 0037              IN(1,LOADER) $                                                 3690          0057
 0038              PTL=0 $                                                                      0058
 0039              SS=0 $                                                                       0059
 0040        STA4.                                                                5360          0060
 0040              ISTATS=V(OFF) $                                                              0061
 0041              JSTATS=V(OFF) $                                                              0062
 0042              IO100(0,LOADER,0,TRIPLT,2) $                                                 0063
 0043              IF LOADER EQ V(EOF) $                                                        0064
 0044              GOTO STA8 $                                                                  0065
 0045              DIRECT                                                                       0066
 0046              LDA     TRIPLT                                                               0067
 0047              LDQ     0,DL                                                                 0068
 0048              LRL     18                                                                   0069
 0049              QRL     18                                                                   0070
 0050              STA     III                                                                  0071
 0051              STQ     JJJ                                                                  0072
 0052              LDA     INSTO                                                                0073
 0053              STA     LOC1                                                                 0074
 0054              STA     LOC0                                                                 0075
 0055              LDA     0,DL                                                                 0076
 0056              LDQ     III                                                                  0077
 0057              TZE     *+4                                                                  0078
 0058              DIV     36,DL                                                                0079
 0059              ALS     18                                                                   0080
 0060              ASA     LOC0                                                                 0081
 0061              LDA     FLAGS,QL                                                             0082
 0062        LOC0  ZERO    0,0                                                                  0083
 0063              TPL     *+3                                                                  0084
 0064              LDA     1,DL                                                                 0085
 0065              STA     ISTATS                                                               0086
 0066              LDA     0,DL                                                                 0087
 0067              LDQ     JJJ                                                                  0088
 0068              DIV     36,DL                                                                0089
 0069              ALS     18                                                                   0090
 0070              ASA     LOC1                                                                 0091
 0071              LDA     FLAGS,QL                                                             0092
 0072        LOC1  ZERO    0,0                                                                  0093
 0073              TPL     STA3                                                                 0094
 0074              LDA     1,DL                                                                 0095
 0075              STA     JSTATS                                                               0096
 0076              JOVIAL                                                                       0097
 0077              IF ISTATS EQ V(ON) AND JSTATS EQ V(ON) $                                     0098
 0078              GOTO STA4 $                                                                  0099
```

Figure V.26. (Continued)

```
2733 01  04-29-71   18.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0079      STA3.                                                                          0100
 0079            RR=0 $                                                      5370         0101
 0080            IFLAG=V(TRUE) $                                             5380         0102
 0081            JFLAG=V(TRUE) $                                             5390         0103
 0082      STA5.                                                             5420         0104
 0082            IF RR EQ PTL $                                              5430         0105
 0083            GOTO STA6 $                                                 5440         0106
 0084            IF DS($RR$) EQ III $                                                     0107
 0085                 BEGIN                                                               0108
 0086                 IFLAG=V(FALSE) $                                                    0109
 0087                 GOTO STA12 $                                                        0110
 0088                 END                                                                 0111
 0089            IF DS($RR$) EQ JJJ $                                                     0112
 0090                 BEGIN                                                  5480         0113
 0091                 JFLAG=V(FALSE) $                                       5490         0114
 0092                 RJ=RR $                                                5500         0115
 0093                 END                                                    5510         0116
 0094      STA12.                                                                         0117
 0094            IF IFLAG EQ V(TRUE) OF JFLAG EQ V(TRUE) $                   5520         0118
 0095                 BEGIN                                                  5530         0119
 0096                 RR=RR+1 $                                              5540         0120
 0097                 GOTO STA5 $                                            5550         0121
 0098                 END                                                    5560         0122
 0099            DS($RJ$)=-1 $                                                            0123
 0100            SS=SS-1 $                                                                0124
 0101            XXX=JJJ $                                                                0125
 0102            GOTO FLAGIT $                                                            0126
 0103            GOTO STA4 $                                                              0127
 0104      STA6.                                                             5570         0128
 0104            IF IFLAG NQ JFLAG $                                         5580         0129
 0105                 BEGIN                                                  5590         0130
 0106                 IF IFLAG EQ V(FALSE) AND JSTATS EQ V(OFF) $                         0131
 0107                      BEGIN                                                          0132
 0108                      XXX=JJJ $                                                      0133
 0109                      GOTO FLAGIT $                                                  0134
 0110                      GOTO STA4 $                                                    0135
 0111                      END                                                            0136
 0112                 IF JFLAG EQ V(FALSE) AND ISTATS EQ V(OFF) $                         0137
 0113                      BEGIN                                                          0138
 0114                      XXX=III $                                                      0139
 0115                      GOTO FLAGIT $                                                  0140
 0116                      GOTO STA4 $                                                    0141
 0117                      END                                                            0142
 0118                 GOTO STA4 $                                                         0143
 0119                 END                                                                 0144
 0120            IF ISTATS EQ V(ON) $                                                     0145
 0121            GOTO STA10 $                                                             0146
 0122            DS($PTL$)=III $                                                          0147
 0123            SS=SS+1 $                                                   5724         0148
 0124            PTL=PTL+1 $                                                 5726         0149
```

Figure V. 26. (Continued)

```
12735 01  04-29-71   18.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0125             IF PTL EQ RANGE $                                        5730        0150
 0126                    BEGIN                                             5731        0151
 0127                    GOTO COMPAC $                                     5733        0152
 0128                    PTL=SS $                                          5735        0153
 0129                    END                                               5737        0154
 0130             IF JSTATS EQ V(OFF) $                                                0155
 0131                    BEGIN                                                         0156
 0132                    XXX=JJJ $                                                     0157
 0133                    GOTO FLAGIT $                                                 0158
 0134                    END                                                           0159
 0135             GOTO STA11 $                                                         0160
 0136        STA10.                                                        5740        0161
 0136             DS($PTL$)=JJJ $                                                      0162
 0137             SS=SS+1 $                                                5760        0163
 0138             PTL=PTL+1 $                                              5762        0164
 0139             IF PTL EQ RANGE $                                        5764        0165
 0140                    BEGIN                                             5766        0166
 0141                    GOTO COMPAC $                                     5768        0167
 0142                    PTL=SS $                                          5770        0168
 0143                    END                                               5772        0169
 0144        STA11.                                                        5840        0170
 0144             IF SS LS NODIST $                                        5850        0171
 0145             GOTO STA4 $                                                          0172
 0146        STA8.                                                         5870        0173
 0146             PDUMP(SS,SS,1) $                                                     0174
 0147             PDUMP(DS($0$),DS($RANGE-1$),0) $                                     0175
 0148             GOTO COMPAC $                                            5880        0176
 0149             GOTO SORTDS $                                            5900        0177
 0150             FOR I=0,1,SS-1 $                                         5910        0178
 0151                    BEGIN                                             5920        0179
 0152                    IO500(4,DSTABL,0,DS($I$),1) $                     5930        0180
 0153                    WAIT(IO500) $                                     5940        0181
 0154                    END                                               5950        0182
 0155             OUT(2,PRNT) $                                            5960        0183
 0156             OUT(2,DSTABL) $                                          5962        0184
 0157             IN(2,LOADER) $                                           5963        0185
 0158             PDUMP $                                                              0186
 0159             STOP $                                                   5966        0187
 0159        ''*************************************************************''       0188
 0159        ''                           FLAGIT                            ''       0189
 0159        ''*************************************************************''       0190
 0160        CLOSE FLAGIT $                                                            0191
 0161        BEGIN                                                                     0192
 0162             DIRECT                                                   5784        0193
 0163        LOC5 LDA       INSTO                                          5786        0194
 0164             STA       LOC2                                           5788        0195
 0165             STA       LOC3                                           5792        0196
 0166             LDA       0,DL                                           5793        0197
 0167             LDQ       XXX                                            5796        0198
 0168             TZE       *+3                                            5798        0199
```

Figure V. 26. (Continued)

```
12735 01  04-29-71   18.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0169            DIV     36,DL                                                        5800           0200
  0170            ALS     18                                                           5802           0201
  0171            ADA     1,DU                                                         5804           0202
  0172            ASA     LOC2                                                         5805           0203
  0173            SBA     36,DU                                                        5806           0204
  0174            NEG                                                                  5808           0205
  0175            ASA     LOC3                                                         5810           0206
  0176            LDA     FLAGS,QL                                                     5812           0207
  0177      LOC2  ZERO    0,C                                                          5814           0208
  0178            ORA     000001,DL                                                    5816           0209
  0179      LOC3  ZERO    0,0                                                          5818           0210
  0180            STA     FLAGS,QL                                                     5820           0211
  0181            JOVIAL                                                               5830           0212
  0182      END                                                                                       0213
  0182      ''***********************************************************''            800            0214
  0182      ''                            COMPAC                          ''            801            0215
  0182      ''***********************************************************''            810            0216
  0183      CLOSE COMPAC $                                                             812            0217
  0184      BEGIN                                                                      820            0218
  0185            COMTRM=0 $                                                           850            0219
  0186            FOR I=0,1,RANGE-1 $                                                  851            0220
  0187                    BEGIN                                                        860            0221
  0188                    IF DS($I$) EQ -1 $                                           870            0222
  0189                    TEST I $                                                     880            0223
  0190                    DS($COMTRM$)=DS($I$) $                                       890            0224
  0191                    COMTRM=COMTRM+1 $                                            900            0225
  0192                    END                                                          910            0226
  0193            SS=COMTRM $                                                          920            0227
  0194            FOR I=COMTRM,1,RANGE-1 $                                             930            0228
  0195                    BEGIN                                                        940            0229
  0196                    DS($I$)=-1 $                                                 950            0230
  0197                    END                                                          960            0231
  0198            END  ''COMPAC''                                                      970            0232
  0198      ''***********************************************************''            6290           0233
  0198      ''                            SORTDS                          ''            6300           0234
  0198      ''***********************************************************''            6310           0235
  0199      CLOSE SORTDS $                                                             6320           0236
  0200      BEGIN                                                                      6330           0237
  0201      SRTDS0.                                                                    6340           0238
  0201            LL=2 $                                                               6350           0239
  0202      SRTDS1.                                                                    6360           0240
  0202            IF LL GQ SS $                                                        6370           0241
  0203            GOTO SRTDS2 $                                                        6380           0242
  0204            LL=LL*2 $                                                            6390           0243
  0205            GOTO SRTDS1 $                                                        6400           0244
  0206      SRTDS2.                                                                    6405           0245
  0206            DD=(LL/2)-1 $                                                        6410           0246
  0207      SRTDS3.                                                                    6415           0247
  0209            II=0 $  JJ=0 $  KK=0 $                                               6420           0248
  0210            DIRECT                                                               6430           0249
```

Figure V.26. (Continued)

```
12735 01  04-29-71   16.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0211       SRTDS4 LDA     II                                       6440      0250
  0212              EAX5    DSTAB+1,AL                               6450      0251
  0213              ADA     DD                                       6460      0252
  0214              EAX4    DSTAB+1,AL                               6470      0253
  0215              LDQ     0,4                                      6480      0254
  0216              CMPQ    0,5                                      6490      0255
  0217              TNC     SRTDS7                                   6500      0256
  0218              JOVIAL                                           6510      0257
  0219       SRTDS5.                                                 6520      0258
  0219              IF II+DD LQ SS $                                 6530      0259
  0220                      BEGIN                                    6540      0260
  0221                      II=II+1 $                                6550      0261
  0222                      GOTO SRTDS4 $                            6560      0262
  0223                      END                                      6570      0263
  0224              DD=(DD-1)/2 $                                    6580      0264
  0225              IF DD LQ 0 $                                     6590      0265
  0226              RETURN $                                         6600      0266
  0227              GOTO SRTDS3 $                                    6610      0267
  0228              DIRECT                                           6620      0268
  0229       SRTDS7 LDA     0,4                                      6630      0269
  0230              LDQ     0,5                                      6640      0270
  0231              LLR     36                                       6650      0271
  0232              STA     0,4                                      6660      0272
  0233              STQ     0,5                                      6670      0273
  0234              JOVIAL                                           6680      0274
  0235              JJ=II-DD $                                       6690      0275
  0236              IF JJ LS 0 $                                     6700      0276
  0237              GOTO SRTDS5 $                                    6710      0277
  0238              KK=II $                                          6720      0278
  0239              DIRECT                                           6730      0279
  0240       SRTDS8 LDA     JJ                                       6740      0280
  0241              EAX5    DSTAB+1,AL                               6750      0281
  0242              LDA     KK                                       6760      0282
  0243              EAX4    DSTAB+1,AL                               6770      0283
  0244              LDQ     0,4                                      6780      0284
  0245              CMPQ    0,5                                      6790      0285
  0246              TZE     SRTDS5                                   6800      0286
  0247              TRC     SRTDS5                                   6810      0287
  0248              LDA     0,4                                      6820      0288
  0249              LDQ     0,5                                      6830      0289
  0250              LLR     36                                       6840      0290
  0251              STA     0,4                                      6850      0291
  0252              STQ     0,5                                      6860      0292
  0253              JOVIAL                                           6870      0293
  0254              JJ=JJ-DD $                                       6875      0294
  0255              IF JJ LS 0 $                                     6880      0295
  0256              GOTO SRTDS5 $                                    6885      0296
  0257              KK=JJ+DD $                                       6890      0297
  0258              GOTO SRTDS8 $                                    6900      0298
  0259       END '' OF DS-TABLE SORTING ROUTINE''                    6920      0299
```

Figure V. 26. (Continued)

```
12735 01  04-2[illegible]   18.712   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS
JOVIAL VERSION -- NUMBER 46
 0260      TEST PROCESS $                                                  11750                 0300
```

Figure V.26. (Concluded)

```
12734 01  04-29-71   16.601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM CLUSTR.2
  0000       START
  0000       ''*************************************************************''
  0000       ''*                         DEFINES                           *''
  0000       ''*************************************************************''
  0001            DEFINE NODIST ''1500'' $ ''NO. OF DISTINCT STEMS''
  0002            DEFINE MAXLEN ''5000'' $ ''NO. OF DICTIONARY STEMS''
  0003            DEFINE MAXCOR ''52500'' $ ''NO. OF WORDS FOR CONC. STORAGE''
  0004            DEFINE CLUSIZ ''4'' $ ''CLUSTER SIZE''
  0005            DEFINE VECSIZ ''6'' $ ''CLUSIZ+2''
  0006            DEFINE UNSINT ''I 35 U'' $
  0007            DEFINE SGNINT ''I 35 S'' $
  0007       ''************************************************************''
  0007       ''                         TABLES                             ''
  0007       ''************************************************************''
  0008            TABLE ALOCOR R MAXCOR S 1 $
  0008                 BEGIN
  0009                 ITEM ALOCO H 6 0 0 $
  0010                 END
  0010            TABLE CORTAB R NODIST S 2 $
  0010                 BEGIN
  0011                 ITEM CORCEN UNSINT 0 0 $
  0012                 ITEM CORREL F      1 0 $
  0013                 END
  0013            TABLE CLUSTR R CLUSIZ S 1 $
  0013                 BEGIN
  0014                 ITEM CLUCEN I 18 S    0  0 $
  0015                 ITEM CLUCOR A 18 S 14 0 18 $
  0016                 END
  0016            TABLE TEMPO R 2 S 1 $
  0016                 BEGIN
  0017                 ITEM DUMMYO H 6 0 0 $
  0018                 END
  0018       ''*************************************************************''
  0018       ''                         ITEMS                               ''
  0018       ''*************************************************************''
  0018            ITEM INDEX UNSINT P 0 $
  0019            ITEM TMPMAG F $
  0020            ITEM DUMMYO SGNINT P 0 $
  0021            ITEM CC F $
  0022            ITEM ICOUNT UNSINT P 0 $
  0023            ITEM JCOUNT UNSINT P 0 $
  0024            ITEM DOTPRD UNSINT P 0 $
  0025            ITEM DOC1 UNSINT P 0 $
  0026            ITEM DOC2 UNSINT P 0 $
  0027            ITEM WT1 UNSINT P 0 $
  0028            ITEM WT2 UNSINT P 0 $
  0028       ''*************************************************************''
  0028       ''                         ARRAYS                              ''
  0028       ''*************************************************************''
```

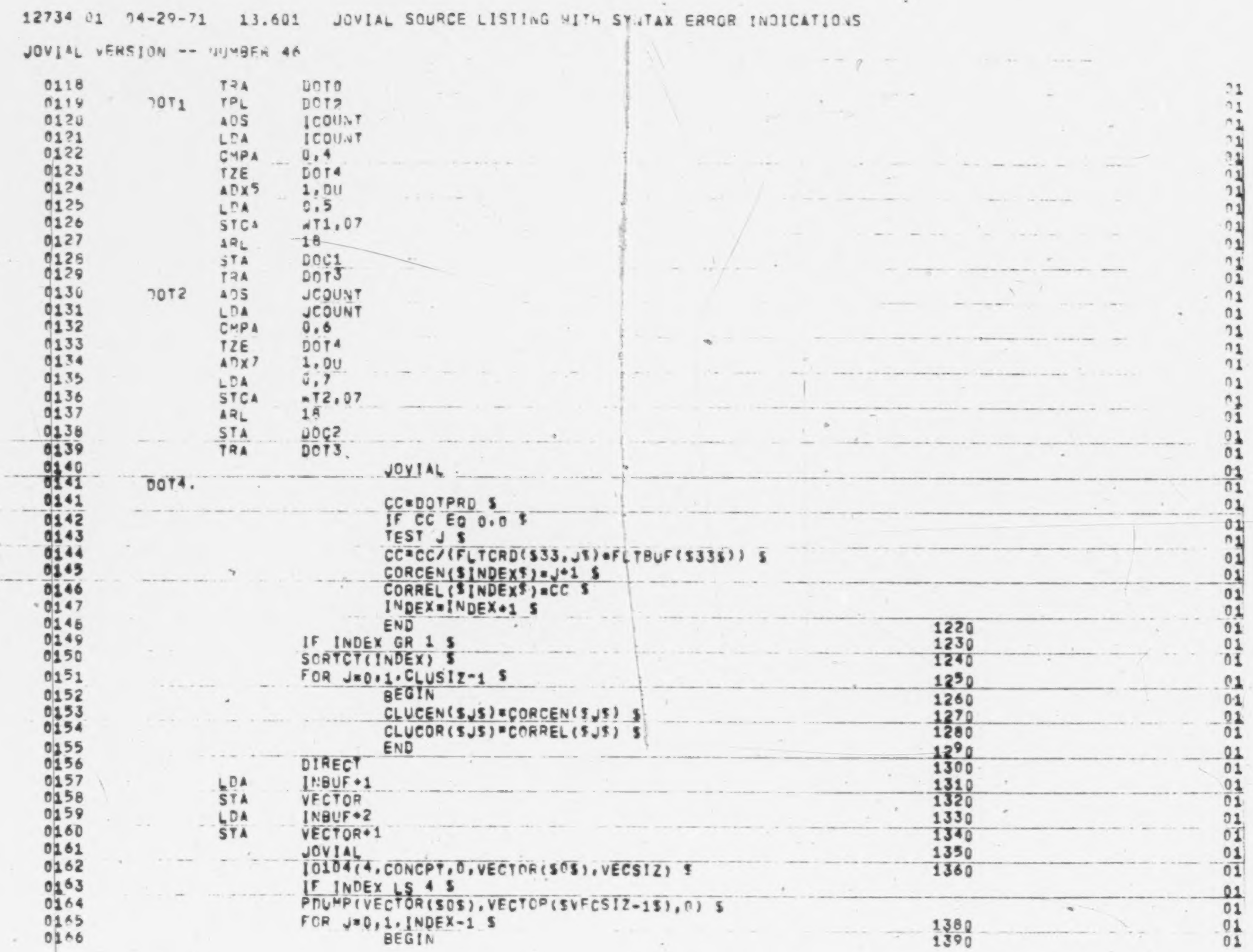
Figure V.27. CLUSTR.2 Listing

```
12734 01  04-29-71   18.601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0029          ARRAY DISTCT NODIST UNSINT $          BEGIN END                 430
 0030          ARRAY INBUF       35 SGNINT $         BEGIN END                 440
 0031          ARRAY FLTBUF      35       F $        BEGIN END                 450
 0032          ARRAY CONCRD      35        1 SGNINT $   BEGIN END              460
 0033          ARRAY FLTCRD      35        1       F $   BEGIN END             470
 0034          ARRAY VECTOR VECSIZ SGNINT $          BEGIN END                 480
 0034.    ''**********************************************************''       490
 0034     ''                        OVERLAYS                            ''       500
 0034     ''**********************************************************''       510
 0035          OVERLAY INBUF =FLTBUF $                                         530
 0036          OVERLAY VECTOR=DUMMYG,CLUSTR $                                  540
 0037          OVERLAY CONCRD=FLTCRD $                                         545
 0037     ''**********************************************************''       550
 0037     ''                        FILES                               ''       560
 0037     ''**********************************************************''       570
 0038          FILE CENTER ''INPUT OF DISTINCT STEM NUMBERS''                  580
 0038               V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R10 $          590
 0039          FILE LOADER ''INPUT OF CONCORDANCES''                           600
 0039               V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R11 $          610
 0040          FILE CONCPT ''OUTPUT OF CLUSTERS''                              620
 0040               V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R12 $          630
 0041     GO.                                                                  640
 0041          OUT(1,CONCPT) $                                                 650
 0042          IN(1,LOADER) $                                                  660
 0043          IN(1,CENTER) $                                                  670
 0044          FOR I=0,1,NODIST-1 $                                            680
 0045               BEGIN                                                      690
 0046               IO100(0,CENTER,0,DISTCT($I$),1) $                          700
 0047               WAIT(IO100) $                                              710
 0048               END                                                        720
 0049          INDEX=0 $                                                       730
 0050          FOR I=0,1,MAXLEN-1 $                                            740
 0051               BEGIN                                                      750
 0052               IO101(0,LOADER,0,INBUF($0$),33) $                          760
 0053               FOR J=0,1,NODIST-1 $                                       770
 0054                    BEGIN                                                 780
 0055                    IF DISTCT($J$) LS I $                                 790
 0056                    TEST J $                                              800
 0057                    IF DISTCT($J$) EQ I $                                 810
 0058                         BEGIN                                            820
 0059                         FOR K=0,1,INBUF($0$)+2 $                         830
 0060                              BEGIN                                       840
 0061                              CONCRD($K,INDEX$)=INBUF($K$) $              850
 0062                              END                                         860
 0063                         TMPMAG=MAG(CONCRD($0,INDEX$)) $                  870
 0064                         FLTCRD($34,INDEX$)=TMPMAG $                      880
 0065                         FLTCRD($33,INDEX$)=TMPMAG**.5 $                  890
 0066                         INDEX=INDEX+1 $                                  900
 0067                         IF INDEX EQ NODIST $                             910
 0068                         GOTO STA1 $                                      920
```

Figure V.27. (Continued)

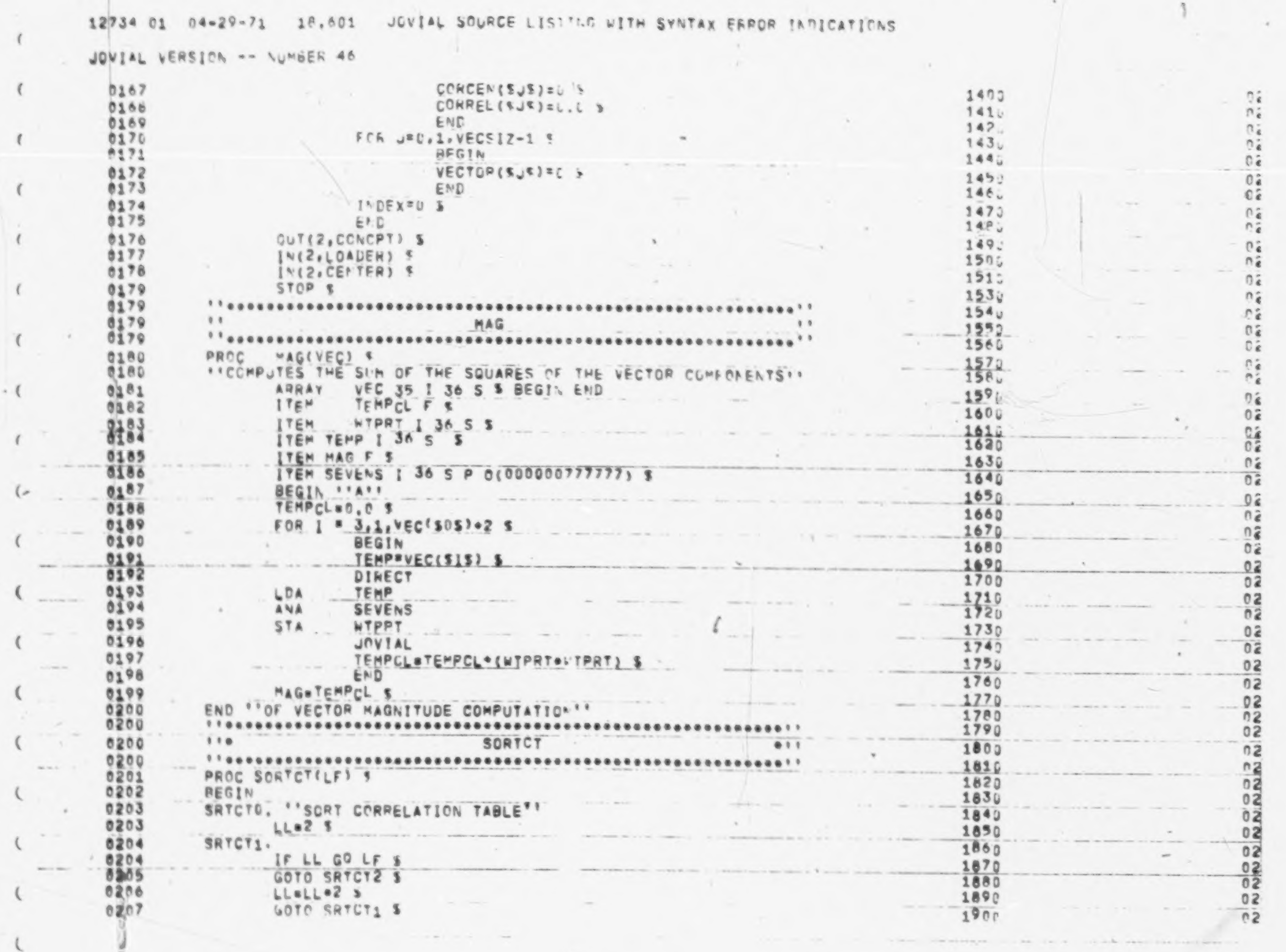
```
12734 01  04-29-71   18.601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0069                                   END                                       930        01
 0070                               TEST I $                                      940        01
 0071                               END                                           950        01
 0072                     TEST I $                                                960        01
 0073                     END                                                     970        01
 0074      STA1.                                                                  980        01
 0074           INDEX=0 $                                                         990        01
 0075           IO102(13,LOADER,0) $                                             1000        01
 0076           FOR I=0,1,MAXLEN-1 $                                             1010        01
 0077                BEGIN                                                       1020        01
 0078                IO103(0,LOADER,0,INBUF($0$),33) $                           1030        01
 0079                TMPMAG=MAG(INBUF($0$)) $                                    1040        01
 0080                FLTBUF($34$)=TMPMAG $                                       1050        01
 0081                FLTBUF($33$)=TMPMAG**.5 $                                   1060        01
 0082                FOR J=0,1,NODIST-1 $                                        1070        01
 0083                     BEGIN                                                  1080        01
 0084                     DIRECT                                                            01
 0085           LDQ     0                                                                   01
 0086           MPY     35,DL                                                               01
 0087           EAX4    CONCRD,QL                                                           01
 0088           EAX5    CONCRD+3,QL                                                         01
 0089           EAX6    INBUF                                                               01
 0090           EAX7    INBUF+3                                                             01
 0091           STZ     ICOUNT                                                              01
 0092           STZ     JCOUNT                                                              01
 0093           STZ     DOTPRD                                                              01
 0094      DOT0 LDA     ICOUNT                                                              01
 0095           CMPA    0,4                                                                 01
 0096           TZE     DOT4                                                                01
 0097           LDA     JCOUNT                                                              01
 0098           CMPA    0,6                                                                 01
 0099           TZE     DOT4                                                                01
 0100           LDA     0,5                                                                 01
 0101           STCA    WT1,07                                                              01
 0102           ARL     18                                                                  01
 0103           STA     DOC1                                                                01
 0104           LDA     0,7                                                                 01
 0105           STCA    WT2,07                                                              01
 0106           ARL     18                                                                  01
 0107           STA     DOC2                                                                01
 0108      DOT3 LDA     DOC1                                                                01
 0109           CMPA    DOC2                                                                01
 0110           TNZ     DOT1                                                                01
 0111           LDQ     WT1                                                                 01
 0112           MPY     WT2                                                                 01
 0113           ASQ     DOTPRD                                                              01
 0114           ADX5    1,DU                                                                01
 0115           ADX7    1,DU                                                                01
 0116           AOS     ICOUNT                                                              01
 0117           AOS     JCOUNT                                                              01
```

Figure V.27. (Continued)



```
12734 01  04-29-71   13.601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0118                 TRA     DOT0                                                              01
  0119        DOT1     TPL     DOT2                                                              01
  0120                 ADS     ICOUNT                                                            01
  0121                 LDA     ICOUNT                                                            01
  0122                 CMPA    0,4                                                               01
  0123                 TZE     DOT4                                                              01
  0124                 ADX5    1,DU                                                              01
  0125                 LDA     0,5                                                               01
  0126                 STCA    *T1,07                                                            01
  0127                 ARL     18                                                                01
  0128                 STA     DOC1                                                              01
  0129                 TRA     DOT3                                                              01
  0130        DOT2     ADS     JCOUNT                                                            01
  0131                 LDA     JCOUNT                                                            01
  0132                 CMPA    0,6                                                               01
  0133                 TZE     DOT4                                                              01
  0134                 ADX7    1,DU                                                              01
  0135                 LDA     0,7                                                               01
  0136                 STCA    *T2,07                                                            01
  0137                 ARL     18                                                                01
  0138                 STA     DOC2                                                              01
  0139                 TRA     DOT3.                                                             01
  0140                                 JOVIAL                                                    01
  0141        DOT4.                                                                              01
  0141                                 CC=DOTPRD $                                               01
  0142                                 IF CC EQ 0.0 $                                            01
  0143                                 TEST J $                                                  01
  0144                                 CC=CC/(FLTCRD($33,J$)*FLTBUF($33$)) $                     01
  0145                                 CORCEN($INDEX$)=J+1 $                                     01
  0146                                 CORREL($INDEX$)=CC $                                      01
  0147                                 INDEX=INDEX+1 $                                           01
  0148                                 END                                         1220          01
  0149                         IF INDEX GR 1 $                                     1230          01
  0150                         SORTCT(INDEX) $                                     1240          01
  0151                         FOR J=0,1,CLUSIZ-1 $                                1250          01
  0152                                 BEGIN                                       1260          01
  0153                                 CLUCEN($J$)=CORCEN($J$) $                   1270          01
  0154                                 CLUCOR($J$)=CORREL($J$) $                   1280          01
  0155                                 END                                         1290          01
  0156                         DIRECT                                              1300          01
  0157                 LDA     INBUF+1                                             1310          01
  0158                 STA     VECTOR                                              1320          01
  0159                 LDA     INBUF+2                                             1330          01
  0160                 STA     VECTOR+1                                            1340          01
  0161                         JOVIAL                                              1350          01
  0162                         IO104(4,CONCPT,0,VECTOR($0$),VECSIZ) $              1360          01
  0163                         IF INDEX LS 4 $                                                   01
  0164                         PDUMP(VECTOR($0$),VECTOR($VECSIZ-1$),0) $                         01
  0165                         FOR J=0,1,INDEX-1 $                                 1380          01
  0166                                 BEGIN                                       1390          01
```

Figure V.27. (Continued)



```
12734 01  04-29-71   18,601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0167                          CONCEN($J$)=0 $                                      1400          02
  0168                          CORREL($J$)=0.0 $                                    1410          02
  0169                          END                                                  1420          02
  0170                 FOR J=0,1,VECSIZ-1 $                                          1430          02
  0171                          BEGIN                                                1440          02
  0172                          VECTOR($J$)=0 $                                      1450          02
  0173                          END                                                  1460          02
  0174                 INDEX=0 $                                                     1470          02
  0175                 END                                                           1480          02
  0176         OUT(2,CONCPT) $                                                       1490          02
  0177         IN(2,LOADER) $                                                        1500          02
  0178         IN(2,CENTER) $                                                        1510          02
  0179         STOP $                                                                1530          02
  0179     ''***********************************************************''           1540          02
  0179     ''                          MAG                              ''           1550          02
  0179     ''***********************************************************''           1560          02
  0180     PROC    MAG(VEC) $                                                        1570          02
  0180     ''COMPUTES THE SUM OF THE SQUARES OF THE VECTOR COMPONENTS''              1580          02
  0181         ARRAY   VEC 35 I 36 S $ BEGIN END                                     1590          02
  0182         ITEM    TEMPCL F $                                                    1600          02
  0183         ITEM    WTPRT I 36 S $                                                1610          02
  0184         ITEM TEMP I 36 S  $                                                   1620          02
  0185         ITEM MAG F $                                                          1630          02
  0186         ITEM SEVENS I 36 S P O(000000777777) $                                1640          02
  0187         BEGIN ''A''                                                           1650          02
  0188         TEMPCL=0.0 $                                                          1660          02
  0189         FOR I = 3,1,VEC($0$)+2 $                                              1670          02
  0190                 BEGIN                                                         1680          02
  0191                 TEMP=VEC($I$) $                                               1690          02
  0192                 DIRECT                                                        1700          02
  0193         LDA     TEMP                                                          1710          02
  0194         ANA     SEVENS                                                        1720          02
  0195         STA     WTPPT                                                         1730          02
  0196                 JOVIAL                                                        1740          02
  0197                 TEMPCL=TEMPCL+(WTPRT*WTPRT) $                                 1750          02
  0198                 END                                                           1760          02
  0199         MAG=TEMPCL $                                                          1770          02
  0200     END ''OF VECTOR MAGNITUDE COMPUTATION''                                   1780          02
  0200     ''***********************************************************''           1790          02
  0200     ''*                          SORTCT                          *''           1800          02
  0200     ''***********************************************************''           1810          02
  0201     PROC SORTCT(LF) $                                                         1820          02
  0202     BEGIN                                                                     1830          02
  0203     SRTCT0. ''SORT CORRELATION TABLE''                                        1840          02
  0203         LL=2 $                                                                1850          02
  0204     SRTCT1.                                                                   1860          02
  0204         IF LL GQ LF $                                                         1870          02
  0205         GOTO SRTCT2 $                                                         1880          02
  0206         LL=LL*2 $                                                             1890          02
  0207         GOTO SRTCT1 $                                                         1900          02
```

Figure V.27. (Continued)

```
12734 01  04-29-71   18.601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION --NUMBER 46
  0208      SRTCT2.                                                        1910          02
  0208            DD=(LL/2)-1 $                                            1920          02
  0209      SRTCT3.                                                        1930          02
  0209            II=0 $                                                   1940          02
  0210            JJ=0 $                                                   1950          02
  0211            KK=0 $                                                   1960          02
  0212            DIRECT                                                   1970          02
  0213      SRTCT4 LDA     II                                              1980          02
  0214            ALS     1                                                              02
  0215            EAX7    CORTAB+1,AL                                      2000          02
  0216            ARS     1                                                2010          02
  0217            ADA     JJ                                               2020          02
  0218            ALS     1                                                2030          02
  0219            EAX6    CORTAB+1,AL                                                    02
  0220            FLD     1,7                                                            02
  0221            FCMP    1,6                                                            02
  0222            TMI     SRTCT7                                                         02
  0223            JOVIAL                                                   2080          02
  0224      SRTCT5.                                                        2090          02
  0224            IF II+DD+1 LS LL $                                       2100          02
  0225            BEGIN                                                    2110          02
  0226            DIRECT                                                   2120          02
  0227            AOS     II                                               2130          02
  0228            TRA     SRTCT4                                           2140          02
  0229            JOVIAL                                                   2150          02
  0230            END                                                      2160          02
  0231            DD=(DD-1)/2 $                                            2170          02
  0232            IF DD LQ 0 $                                             2180          02
  0233            RETURN $                                                 2190          02
  0234            GOTO SRTCT3 $                                            2200          02
  0235      SRTCT7.                                                        2210          02
  0235            DIRECT                                                   2220          02
  0236            LDAQ    0,7                                              2230          02
  0237            STAQ    TEMPO+1                                          2240          02
  0238            LDAQ    0,6                                              2250          02
  0239            STAQ    0,7                                              2260          02
  0240            LDAQ    TEMPO+1                                          2270          02
  0241            STAQ    0,6                                              2280          02
  0242            JOVIAL                                                   2290          02
  0243            JJ=II-DD $                                               2300          02
  0244            IF JJ LS 0 $                                             2310          02
  0245            GOTO SRTCT5 $                                            2320          02
  0246            KK=II $                                                  2330          02
  0247            DIRECT                                                   2340          02
  0248      SRTCT8 LDA     JJ                                              2350          02
  0249            ALS     1                                                              02
  0250            EAX5    CORTAB+1,AL                                                    02
  0251            LDA     KK                                               2380          02
  0252            ALS     1                                                              02
  0253            EAX4    CORTAB+1,AL                                                    02
```

Figure V.27. (Continued)

```
12734 01  04-29-71   19.601   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION 3-- NUMBER 46

  0254              FLD     1.5                                                     03
  0255              FCMP    1,4                                                     03
  0256              TZE     SRTCT5                                                  03
  0257              TPL     SRTCT5                                                  03
  0258              LDAQ    0,5                                  2450               03
  0259              STAQ    TEMP0+1                              2460               03
  0260              LDAQ    0,4                                  2470               03
  0261              STAQ    0,5                                  2480               03
  0262              LDAQ    TEMP0+1                              2490               03
  0263              STAQ    0,4                                  2500               03
  0264              JOVIAL                                       2510               03
  0265              JJ=JJ-DD $                                   2520               03
  0266              IF JJ LS 0 $                                 2530               03
  0267              GOTO SRTCT5 $                                2540               03
  0268              KK=JJ*DD $                                   2550               03
  0269              GOTO SRTCT8 $                                2560               03
  0270        END                                                2570               03
  0271        TERM GO $                                          3690               03
```

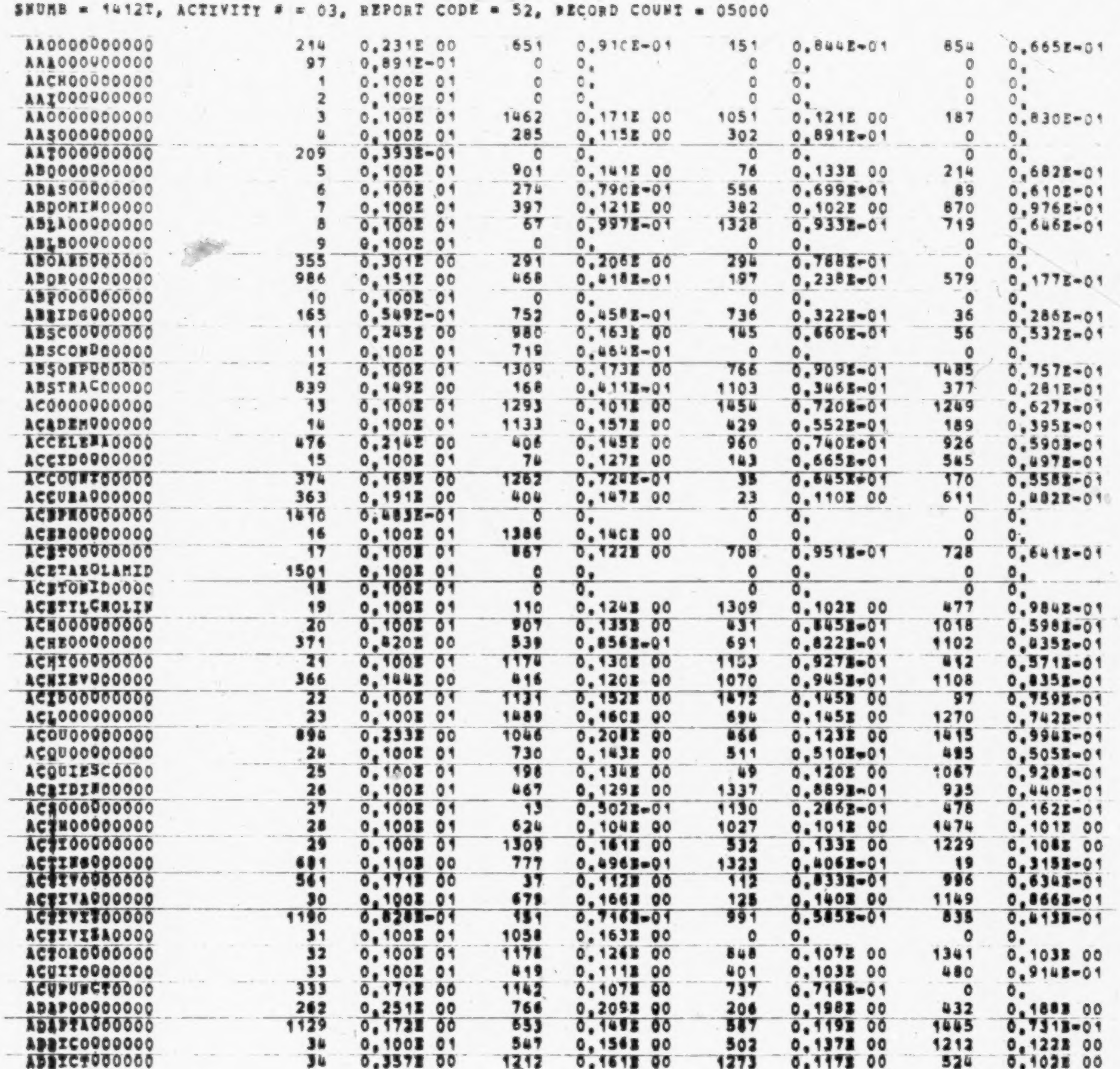
Figure V.27. (Concluded)

V. 6. 3 Summary of Concept-Dictionary Generation

The clustering routines by themselves form a package, along with the statistical filter, which partitions the stems obtained from a document collection into 1501 semantic domains. Figure V. 28 is the first page of the stem-dictionary actually being used with the on-line system and was produced directly by the last phase of the stem clustering package.

Several residual problems remain as food for additional thought. In the first place, during distinct-stem generation the process is stopped when 1500 distinct stems have been isolated. It certainly would result in a lower correlation measure between the individual stems if the process were continued until the peak number of distinct stems is reached. The process would then be stopped on the "down" side of the curve, as the number of distinct stems starts decreasing. The result of this change would be the following: since the source for the distinct-stem computation is the tape-file of cosines in descending order, it stands to reason that the longer the process is allowed to continue the lower the correlation measure between distinct stems, thus resulting in a better separation.

The only other problem of appreciable size is the matter of the amount of CPU time required to perform the clustering process, as well as the large amount of core storage needed. Given the present technology, there is no way to speed up the clustering operation short of utilizing a faster computer to do the job. The amount of storage used could be drastically reduced at the cost of vastly increased execution time through the use of random-access external media.



```
SNUMB = 1412T, ACTIVITY # = 03, REPORT CODE = 52, RECORD COUNT = 05000

AA0000000000      214  0,231E 00   651  0,910E-01   151  0,844E-01   854  0,665E-01
AAA000000000       97  0,891E-01     0  0,            0  0,            0  0,
AACH00000000        1  0.100E 01     0  0.            0  0.            0  0,
AAI000000000        2  0,100E 01     0  0,            0  0,            0  0,
AAO000000000        3  0,100E 01  1462  0,171E 00  1051  0,121E 00   187  0,830E-01
AAS000000000        4  0,100E 01   285  0,115E 00   302  0,891E-01     0  0,
AAT000000000      209  0,393E-01     0  0,            0  0,            0  0,
AB0000000000        5  0,100E 01   901  0,141E 00    76  0,133E 00   214  0,682E-01
ABAS00000000        6  0,100E 01   274  0,790E-01   556  0,699E-01    89  0,610E-01
ABDOMIN00000        7  0,100E 01   397  0,121E 00   382  0,102E 00   870  0,976E-01
ABLA00000000        8  0,100E 01    67  0,997E-01  1328  0,933E-01   719  0,646E-01
ABLE00000000        9  0,100E 01     0  0,            0  0,            0  0,
ABOARD000000      355  0,301E 00   291  0,206E 00   294  0,788E-01     0  0,
ABOR00000000      986  0,151E 00   468  0,418E-01   197  0,238E-01   579  0,177E-01
ABR000000000       10  0,100E 01     0  0.            0  0.            0  0,
ABRIDG000000      165  0,549E-01   752  0,458E-01   736  0,322E-01    36  0,286E-01
ABSC00000000       11  0,245E 00   980  0,163E 00   145  0,660E-01    56  0,532E-01
ABSCOND00000       11  0,100E 01   719  0,464E-01     0  0,            0  0,
ABSORP000000       12  0,100E 01  1309  0,173E 00   766  0,909E-01  1485  0,757E-01
ABSTRAC00000      839  0,149E 00   168  0,411E-01  1103  0,346E-01   377  0,281E-01
AC0000000000       13  0,100E 01  1293  0,101E 00  1454  0,720E-01  1249  0,627E-01
ACADEM000000       14  0,100E 01  1133  0,157E 00   429  0,552E-01   189  0,395E-01
ACCELERA0000      476  0,214E 00   406  0,145E 00   960  0,740E-01   926  0,590E-01
ACCID0000000       15  0,100E 01    74  0,127E 00   143  0,665E-01   545  0,497E-01
ACCOUNT00000      374  0,169E 00  1262  0,724E-01    35  0,645E-01   170  0,558E-01
ACCURA000000      363  0,191E 00   404  0,147E 00    23  0,110E 00   611  0,482E-01
ACEPH0000000     1410  0,483E-01     0  0,            0  0,            0  0,
ACER00000000       16  0,100E 01  1386  0,140E 00     0  0,            0  0,
ACET00000000       17  0,100E 01   867  0,122E 00   708  0,951E-01   728  0,641E-01
ACETAZOLAMID     1501  0,100E 01     0  0,            0  0.            0  0,
ACETONID000C       18  0,100E 01     0  0,            0  0,            0  0,
ACETYLCHOLIN       19  0,100E 01   110  0,124E 00  1309  0,102E 00   477  0,984E-01
ACH000000000       20  0,100E 01   907  0,135E 00   431  0,845E-01  1018  0,598E-01
ACHE00000000      371  0,820E 00   539  0,856E-01   691  0,822E-01  1102  0,435E-01
ACHI00000000       21  0,100E 01  1174  0,130E 00  1153  0,927E-01   412  0,571E-01
ACHIEV000000      366  0,144E 00   416  0,120E 00  1070  0,945E-01  1108  0,835E-01
ACID00000000       22  0,100E 01  1131  0,152E 00  1472  0,145E 00    97  0,759E-01
ACL000000000       23  0,100E 01  1489  0,160E 00   694  0,145E 00  1270  0,742E-01
ACOU00000000      894  0,233E 00  1046  0,208E 00   466  0,123E 00  1415  0,994E-01
ACQU00000000       24  0,100E 01   730  0,143E 00   511  0,510E-01   495  0,505E-01
ACQUIESC0000       25  0,100E 01   198  0,134E 00    49  0,120E 00  1067  0,928E-01
ACRIDIN00000       26  0,100E 01   467  0,129E 00  1337  0,889E-01   935  0,440E-01
ACS000000000       27  0,100E 01    13  0,502E-01  1130  0,286E-01   478  0,162E-01
ACTH00000000       28  0,100E 01   624  0,104E 00  1027  0,101E 00  1474  0,101E 00
ACTI00000000       29  0,100E 01  1309  0,161E 00   532  0,133E 00  1229  0,108E 00
ACTING000000      681  0,110E 00   777  0,496E-01  1323  0,406E-01    19  0,315E-01
ACTIV0000000      561  0,171E 00    37  0,112E 00   112  0,833E-01   996  0,634E-01
ACTIVA000000       30  0,100E 01   679  0,166E 00   125  0,140E 00  1149  0,866E-01
ACTIVIT00000     1190  0,828E-01   151  0,716E-01   991  0,585E-01   838  0,413E-01
ACTIVISA0000       31  0,100E 01  1058  0,163E 00     0  0,            0  0,
ACTOR0000000       32  0,100E 01  1178  0,126E 00   848  0,107E 00  1341  0,103E 00
ACUIT0000000       33  0,100E 01   419  0,111E 00   401  0,103E 00   480  0,914E-01
ACUPUNCT0000      333  0,171E 00  1142  0,107E 00   737  0,718E-01     0  0,
ADAP00000000      262  0,251E 00   766  0,209E 00   206  0,198E 00   432  0,188E 00
ADAPTA000000     1129  0,172E 00   653  0,149E 00   587  0,119E 00  1445  0,731E-01
ADDIC0000000       34  0,100E 01   547  0,156E 00   502  0,137E 00  1212  0,122E 00
ADDICT000000       34  0,357E 00  1212  0,161E 00  1273  0,117E 00   524  0,102E 00
```

Figure V.28. Stem Dictionary Listing

V.7 DOCUMENT CONCEPT VECTOR GENERATION

V.7.1 Techniques

In the operation of the on-line retrieval system each document in the collection must be characterized by a document concept vector, the sum of all the one-to-many weighted mappings of concept stems into concepts. The following paragraphs describe the generation of document concept vectors (DCV) from the stem dictionary and the microconcordance of each of the documents in the collection.

For each document in the collection the microconcordance generator produces an output tape of stems and weights, identified as to the document in which each one occurs. All these stems are not, however, content stems, as these are identified later on in the statistical filter program. However, a dictionary of content stem is available in a compact form, in which all the stems in a single microconcordance may be looked up.

As each stem for a particular document is encountered in the microconcordance, it is looked up the dictionary mentioned above. If it is indeed a content stem, its four pairs of concept-number/weight pairs are added to the growing DCV for this document. The DCV itself is at first constructed as a sparse vector, and, after all the stems for a particular document have been processed, a compressed version is constructed in core to be written on magnetic tape.

As a result of the summation mentioned above (weights are added to the DCV), the number of occurrences of a content stem within a particular document is taken into account. In this way the document concept vector which is generated ultimately reflects in a quite accurate manner the relative importance of different content stems within a document.

V.7.2 DCVGEN: Program Description

Routine DCVGEN, which generates the Document Concept Vectors, is the last major program in the data base characterization phase of the On-Line Retrieval System. For each document in the collection, a concept vector is generated which defines the orientation of that document in the hyperspace. Also, a file of concept/document pairs is generated in order to construct an inverted list of document concept vectors to be used by the retrieval routine of the dialogue processor.

Input to DCVGEN consists of two files:

1) A merged tape of all microconcordances ordered by ascending document number.
2) The Concept Clusters generated by program CLUSTR.2. The two output files are:
   a) The compressed Document Concept Vectors, and
   b) The Concept/Document pairs.

V.7.3 Flowchart Comments - Routine DCVGEN Refers to Figure V.27

1) Open all I/O files
2) Zero-fill the table used to generate the weighted Document Concept Vector, table Vector
3) Fill array CLUSTR with the 5,000 dictionary stem concept clusters.
4) Read a microconcordance from input file MICRON
5) Test for a new document number in the current microconcordance. The old document number, DOCNUM, is preset to 1, and is reset when the microconcordances of the next document are encountered (document number is the third field of each microconcordance).

   DOCNUM NQ INWRD (2)?

   Yes ⇒ Packet No ⇒6
6) Perform a binary search of the CLUSTR array to determine if the stem found in the current microconcordance is indeed a dictionary stem

7) Was the stem found in CLUSTR?

Yes ⇒8 No ⇒4

8) Add to the growing, sparse Document Concept Vector the four (or less) concept numbers and corresponding weights found in the Concept Cluster for the current microconcordance. The mechanics are as follows:

- o Unpack a Concept/Weight pair from the proper Concept Cluster
- o Use the Concept Number to index the sparse vector, table VECTOR
  - a. Enter the concept number in table vector
  - b. Multiply the concept weight by the number of times the stem appears in the document (the Hth field of the microconcordance) and add this value to the present weight of that concept in table VECTOR.
- o Fetch the next Concept/Weight pair.

9) Pack the sparse document concept vector. Store the number of Concept/Weight pairs in stem CWTALY.

10) Construct the header word of the DCV.

Format: bits 0-17 C(DOCNUM)
bits 18-35 C(CWTALY)

11) Write the final Document Concept Vector on output file COMVEC.

12) For each concept found in the current Document Concept Vector, generate a Concept/Document record on output file DNTVEC.

13) Zero-fill the Document Concept Vector buffer.

14) Reset stem DOCNUM

15) EOF MICRON?

Yes ⇒16 No ⇒5

V.7.4 Operation Procedure - Routine DCVGEN

1) Preset Defines:

MAXLEN - the number of dictionary Clusters to be input

HODIST - the number of distinct stems

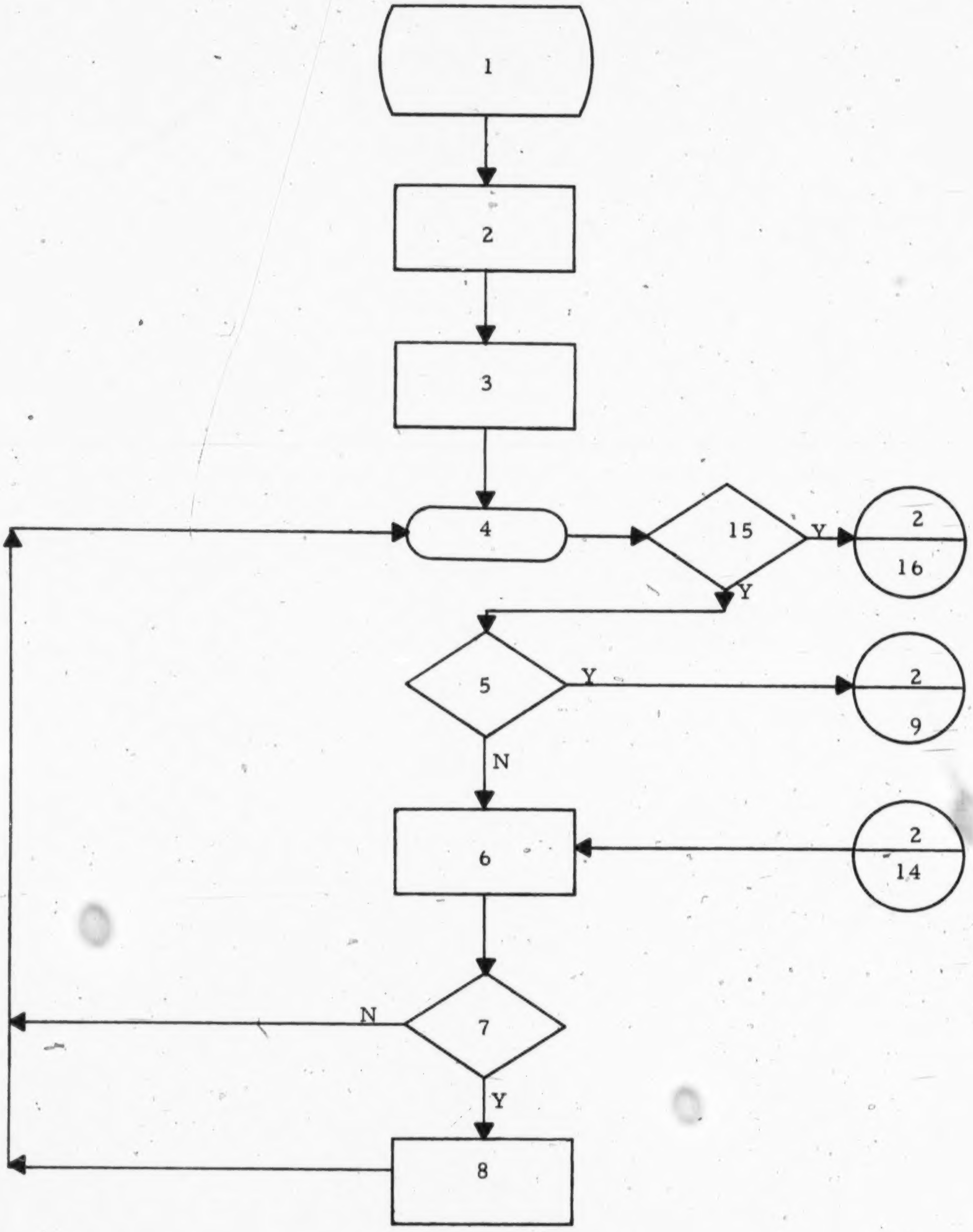

1
2
3
4
15
Y
2
16
Y
5
Y
2
9
N
6
2
14
N
7
Y
8

Figure V.29. DCVGEN Flowchart

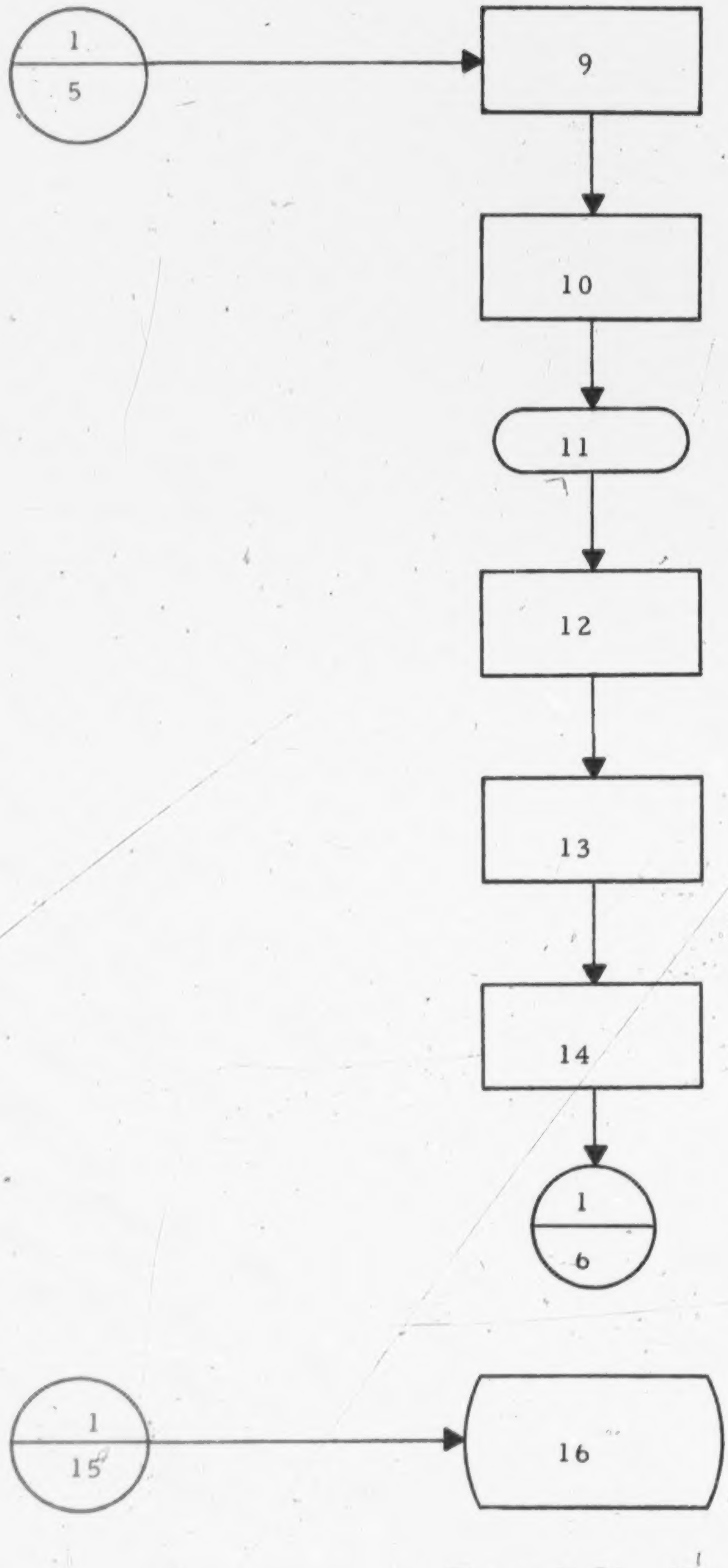
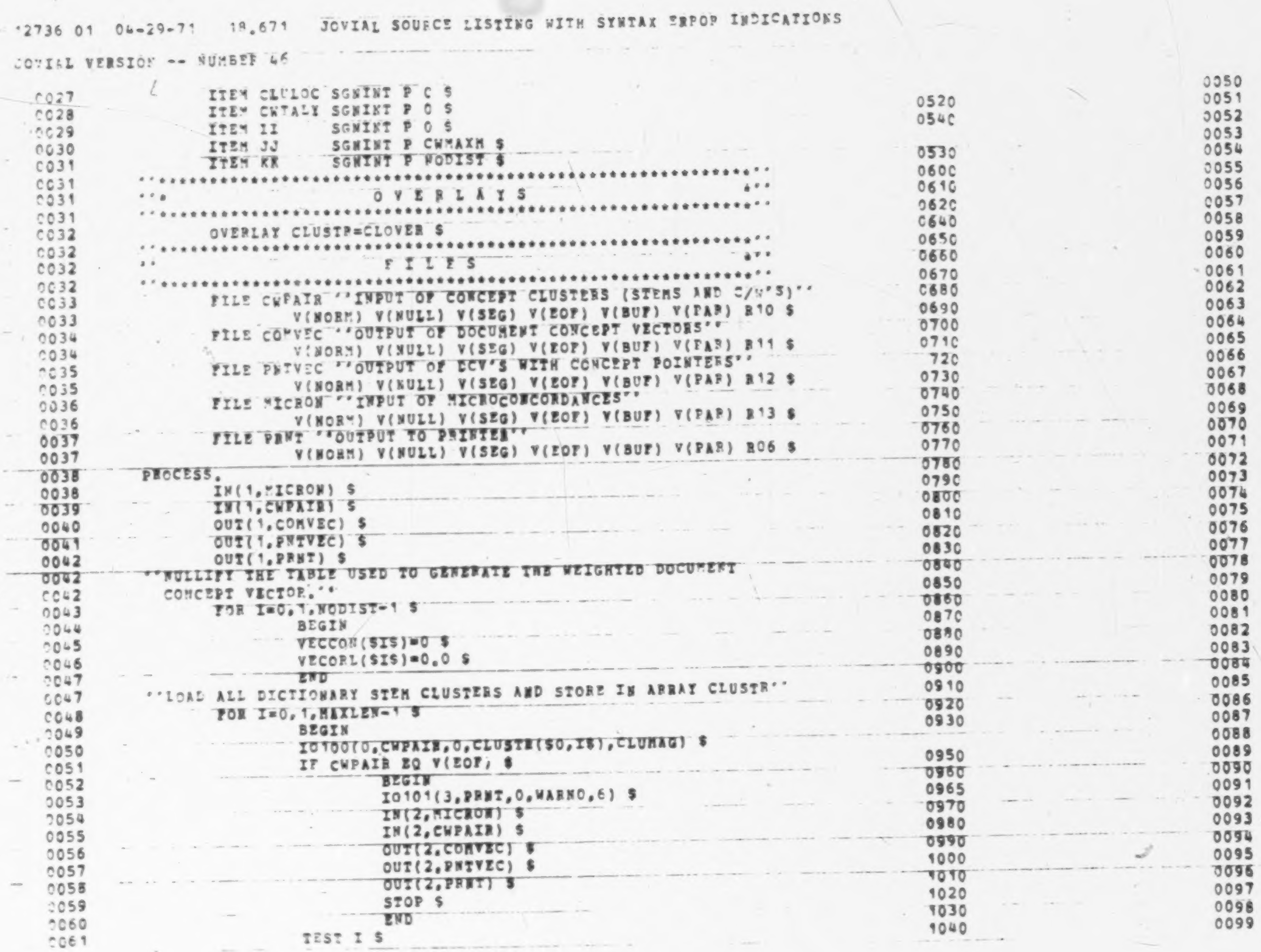

1
5
9
10
11
12
13
14
1
6
1
15
16

Figure V.29. (Concluded)

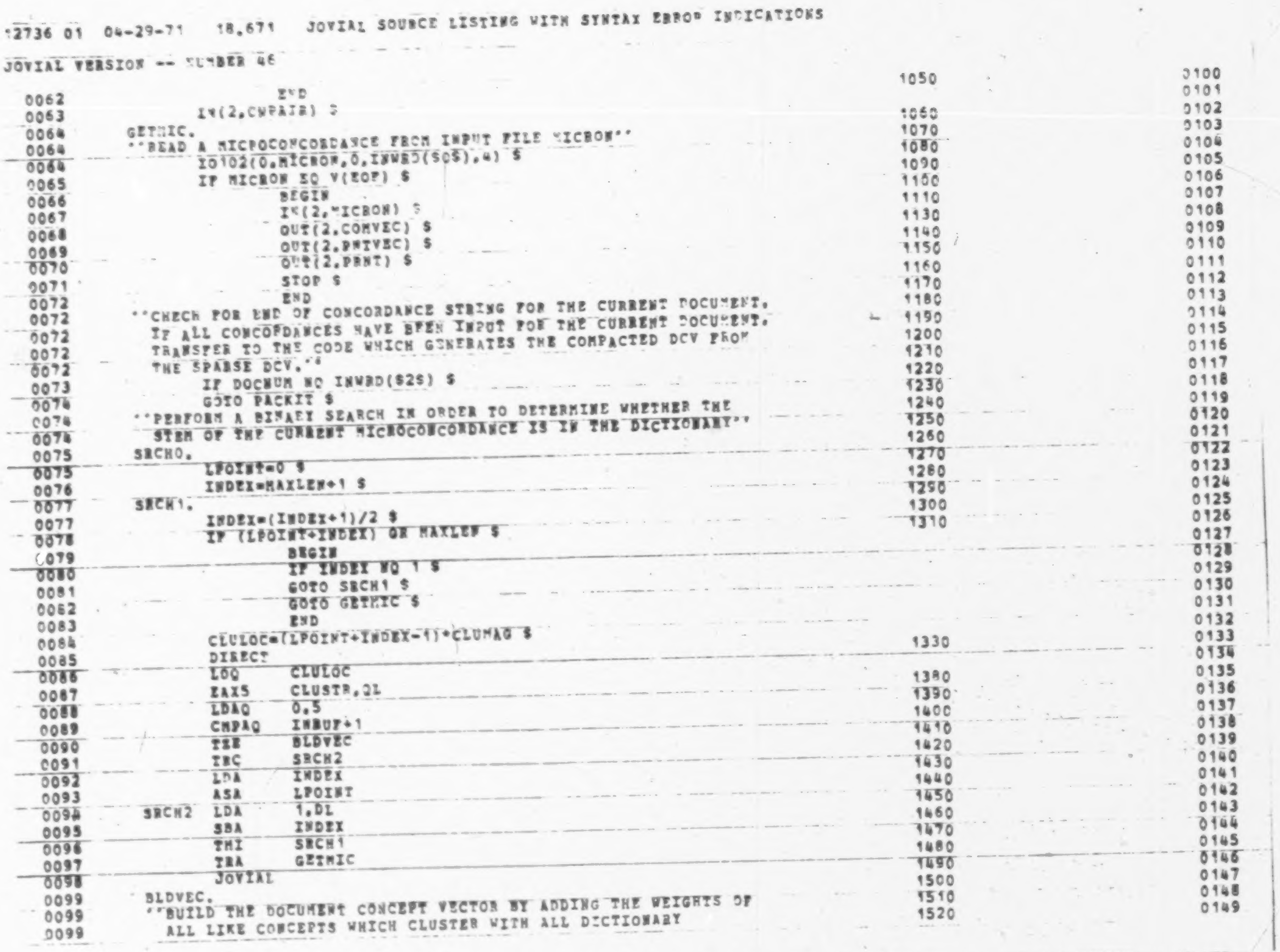
```
12736 01  04-29-71   18.671   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS
JOVIAL VERSION -- NUMBER 46

PROGRAM DCVGEN
  0000      START                                                                         0050          0001
  0000      ''*********************************************************''                0060          0002
  0000      ''*                        D E F I N E S                  *''                0070          0003
  0000      ''*********************************************************''                0080          0004
  0001           DEFINE MAXLEN ''5000'' $ ''NUMBER OF DICTIONARY STEMS''                               0005
  0002           DEFINE NODIST ''1501'' $ ''NUMBER OF DISTINCT STEMS''                                 0006
  0003           DEFINE MAXCOR ''30000'' $ ''MAXLEN X CLUMAG''                                         0007
  0004           DEFINE CLUSIZ ''4'' $ ''NUMBER OF C/W PAIRS PER CLUSTER''                0110          0008
  0005           DEFINE CLUMAG ''6'' $ ''CLUSIZ + TWO WORDS FOR THE STEM''                              0009
  0006           DEFINE DCVSIZ ''202'' $ ''MAX. SIZE OF DCV WITH CONCEPT POINTER''         112          0010
  0007           DEFINE CWMAXM ''200'' $ ''MAX. NO. OF C/W PAIRS PER DCV''                 114          0011
  0008           DEFINE SGNINT ''I 36 S'' $ ''SIGNED, 36-BIT INTEGER''                   0130          0012
  0008      ''*********************************************************''                0140          0013
  0008      ''*                        T A B L E S                    *''                0150          0014
  0008      ''*********************************************************''                0160          0015
  0009           TABLE ALACOR R MAXCOR S 1 $                                              0170          0016
  0009                BEGIN                                                                0180          0017
  0010                ITEM ALOCAT H 6 0 0 $                                                0190          0018
  0011                END                                                                  0200          0019
  0011           TABLE OFECW R 1 S 1 $                                                     210          0020
  0011                BEGIN                                                                0220          0021
  0012                ITEM CONNUM I 18 S    0  0 $                                         0240          0022
  0013                ITEM CONWGT A 18 S 14 0 18 $                                         0250          0023
  0014                END                                                                  0260          0024
  0014           TABLE INBUF R 4 S 1 $                                                     0270          0025
  0014                BEGIN                                                                0280          0026
  0015                ITEM INWRD SGNINT 0 0 $                                              0290          0027
  0016                END                                                                  0300          0028
  0016           TABLE VECTOR R NODIST S 1 $                                               0310          0029
  0016                BEGIN                                                                0320          0030
  0017                ITEM VECCON I 18 S    0  0 $                                         0330          0031
  0018                ITEM VECORL A 18 S 14 0 18 $                                         0340          0032
  0019                END                                                                  0350          0033
  0019      ''*********************************************************''                0360          0034
  0019      ''*                        A R R A Y S                    *''                0370          0035
  0019      ''*********************************************************''                0380          0036
  0019           ARRAY CLUSTR CLUMAG 1 H 6 $                                                            0037
  0020           ARRAY CLOVER CLUMAG 1 F   $                                                            0038
  0021           ARRAY OUTVEC DCVSIZ H 6 $                                                 410          0039
  0021      ''*********************************************************''                0420          0040
  0021      ''*                        I T E M S                      *''                0430          0041
  0021      ''*********************************************************''                0440          0042
  0022           ITEM WARN0 H 36 P                                                         0450          0043
  0022           36H(CLUSTERS INPUT LESS THAN MAXLEN.      ) $                              460          0044
  0023           ITEM WARN1 H 24 P                                                         0470          0045
  0023           24H(NULL VECTOR FOR DOCUMENT) $                                           0480          0046
  0024           ITEM DOCNUM SGNINT P 1 $                                                  0490          0047
  0025           ITEM LPOINT SGNINT P 0 $                                                  0500          0048
  0026           ITEM INDEX  SGNINT P 0 $                                                  0510          0049
```

Figure V.29a. DCVGEN Listing



```
2736 01  04-29-71   18.671   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

0027           ITEM CLULOC SGNINT P C $                                                   0050
0028           ITEM CNTALY SGNINT P 0 $                                           0520    0051
0029           ITEM II     SGNINT P 0 $                                           0540    0052
0030           ITEM JJ     SGNINT P CWMAXM $                                              0053
0031           ITEM KK     SGNINT P NODIST $                                      0530    0054
0031      ''*********************************************************''          0600    0055
0031      ''*                     O V E R L A Y S                   *''          0610    0056
0031      ''*********************************************************''          0620    0057
0032           OVERLAY CLUSTR=CLOVER $                                            0640    0058
0032      ''*********************************************************''          0650    0059
0032      ''*                       F I L E S                       *''          0660    0060
0032      ''*********************************************************''          0670    0061
0033           FILE CWPAIR ''INPUT OF CONCEPT CLUSTERS (STEMS AND C/W'S)''        0680    0062
0033                V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R10 $             0690    0063
0034           FILE COMVEC ''OUTPUT OF DOCUMENT CONCEPT VECTORS''                 0700    0064
0034                V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R11 $             0710    0065
0035           FILE PNTVEC ''OUTPUT OF DCV'S WITH CONCEPT POINTERS''               720    0066
0035                V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R12 $             0730    0067
0036           FILE MICRON ''INPUT OF MICROCONCORDANCES''                         0740    0068
0036                V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R13 $             0750    0069
0037           FILE PRNT ''OUTPUT TO PRINTER''                                    0760    0070
0037                V(NORM) V(NULL) V(SEG) V(EOF) V(BUF) V(PAR) R06 $             0770    0071
0038      PROCESS.                                                                0780    0072
0038           IN(1,MICRON) $                                                     0790    0073
0039           IN(1,CWPAIR) $                                                     0800    0074
0040           OUT(1,COMVEC) $                                                    0810    0075
0041           OUT(1,PNTVEC) $                                                    0820    0076
0042           OUT(1,PRNT) $                                                      0830    0077
0042      ''NULLIFY THE TABLE USED TO GENERATE THE WEIGHTED DOCUMENT               0840    0078
0042        CONCEPT VECTOR.''                                                     0850    0079
0043           FOR I=0,1,NODIST-1 $                                               0860    0080
0044                BEGIN                                                         0870    0081
0045                VECCON($I$)=0 $                                               0880    0082
0046                VECORL($I$)=0.0 $                                             0890    0083
0047                END                                                           0900    0084
0047      ''LOAD ALL DICTIONARY STEM CLUSTERS AND STORE IN ARRAY CLUSTR''          0910    0085
0048           FOR I=0,1,MAXLEN-1 $                                               0920    0086
0049                BEGIN                                                         0930    0087
0050                IO100(0,CWPAIR,0,CLUSTR($0,I$),CLUMAG) $                              0088
0051                IF CWPAIR EQ V(EOF) $                                         0950    0089
0052                     BEGIN                                                    0960    0090
0053                     IO101(3,PRNT,0,WARNO,6) $                                0965    0091
0054                     IN(2,MICRON) $                                           0970    0092
0055                     IN(2,CWPAIR) $                                           0980    0093
0056                     OUT(2,COMVEC) $                                          0990    0094
0057                     OUT(2,PNTVEC) $                                          1000    0095
0058                     OUT(2,PRNT) $                                            1010    0096
0059                     STOP $                                                   1020    0097
0060                     END                                                      1030    0098
0061                TEST I $                                                      1040    0099
```

Figure V.29a. (Continued)



```
12736 01  04-29-71   18,671   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46
                                                                          1050            0100
 0062                   END                                                               0101
 0063           IN(2,CMPAIR) $                                            1060            0102
 0064     GETMIC.                                                         1070            0103
 0064     ''READ A MICROCONCORDANCE FROM INPUT FILE MICRON''              1080            0104
 0064           IO102(0,MICRON,0,INWRD($0$),4) $                          1090            0105
 0065           IF MICRON EQ V(EOF) $                                     1100            0106
 0066                   BEGIN                                             1110            0107
 0067                   IN(2,MICRON) $                                    1130            0108
 0068                   OUT(2,COMVEC) $                                   1140            0109
 0069                   OUT(2,PNTVEC) $                                   1150            0110
 0070                   OUT(2,PRNT) $                                     1160            0111
 0071                   STOP $                                            1170            0112
 0072                   END                                               1180            0113
 0072     ''CHECK FOR END OF CONCORDANCE STRING FOR THE CURRENT DOCUMENT. 1190            0114
 0072       IF ALL CONCORDANCES HAVE BEEN INPUT FOR THE CURRENT DOCUMENT, 1200            0115
 0072       TRANSFER TO THE CODE WHICH GENERATES THE COMPACTED DCV FROM   1210            0116
 0072       THE SPARSE DCV.''                                             1220            0117
 0073            IF DOCNUM NQ INWRD($2$) $                                1230            0118
 0074            GOTO PACKIT $                                            1240            0119
 0074     ''PERFORM A BINARY SEARCH IN ORDER TO DETERMINE WHETHER THE     1250            0120
 0074       STEM OF THE CURRENT MICROCONCORDANCE IS IN THE DICTIONARY''   1260            0121
 0075     SRCHO.                                                          1270            0122
 0075            LPOINT=0 $                                               1280            0123
 0076            INDEX=MAXLEN+1 $                                         1290            0124
 0077     SRCH1.                                                          1300            0125
 0077            INDEX=(INDEX+1)/2 $                                      1310            0126
 0078            IF (LPOINT+INDEX) GR MAXLEN $                                            0127
 0079                     BEGIN                                                           0128
 0080                     IF INDEX NQ 1 $                                                 0129
 0081                     GOTO SRCH1 $                                                    0130
 0082                     GOTO GETMIC $                                                   0131
 0083                     END                                                             0132
 0084            CLULOC=(LPOINT+INDEX-1)*CLUMAG $                         1330            0133
 0085            DIRECT                                                                   0134
 0086            LDQ      CLULOC                                          1380            0135
 0087            EAX5     CLUSTR,QL                                       1390            0136
 0088            LDAQ     0,5                                             1400            0137
 0089            CMPAQ    INBUF+1                                         1410            0138
 0090            TZE      BLDVEC                                          1420            0139
 0091            TRC      SRCH2                                           1430            0140
 0092            LDA      INDEX                                           1440            0141
 0093            ASA      LPOINT                                          1450            0142
 0094      SRCH2 LDA      1,DL                                            1460            0143
 0095            SBA      INDEX                                           1470            0144
 0096            TMI      SRCH1                                           1480            0145
 0097            TRA      GETMIC                                          1490            0146
 0098            JOVIAL                                                   1500            0147
 0099      BLDVEC.                                                        1510            0148
 0099      ''BUILD THE DOCUMENT CONCEPT VECTOR BY ADDING THE WEIGHTS OF   1520            0149
 0099        ALL LIKE CONCEPTS WHICH CLUSTER WITH ALL DICTIONARY
```

Figure V.29a. (Continued)

```
12736 01  04-29-71   18.671   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0099        STEM FOUND IN THE DOCUMENT. BRIEFLY, THIS IS MERELY AN                                   1530          0150
 0099        ADDITION OF THE MAGNITUDES OF EACH COMPONENT OF THE "ECTOR''                             1540          0151
 0099                FOR I=0,1,CLUSIZ-1 $                                                             1550          0152
 0100                        BEGIN                                                                    1560          0153
 0101                        DIRECT                                                                   1570          0154
 0102                LDA     2,5                                                                      1580          0155
 0103                TZE     GETMIC                                                                   1590          0156
 0104                STA     ONECW+1                                                                  1600          0157
 0105                ADX5    1,DU                                                                     1610          0158
 0106                        JOVIAL                                                                   1620          0159
 0107                        VECCON($CONNUM($0$)-1$)=CONNUM($0$) $                                    1630          0160
 0108                        VECOBL($CONNUM($0$)-1$)=                                                 1640          0161
 0108                        VECOBL($CONNUM($0$)-1$)+(CONWGT($0$)*INWRD($3$)) $                                     0162
 0109                        END                                                                      1660          0163
 0110                GOTO GETMIC $                                                                    1670          0164
 0111        PACKIT.                                                                                  1680          0165
 0111        ''THE FOLLOWING CODE PACKS THE SPARSE DOCUMENT CONCEPT VECTOR''                          1690          0166
 0111                DIRECT                                                                           1700          0167
 0112                STZ     II                                                                       1705          0168
 0113                STZ     CNTALY                                                                   1710          0169
 0114                EAX4    OUTVEC+2                                                                 1720          0170
 0115                LDA     II                                                                       1730          0171
 0116        PACK0   LDQ     VECTOR+1,AL                                                              1740          0172
 0117                CMPQ    0,DL                                                                                   0173
 0118                TZE     PACK1                                                                    1760          0174
 0119                LDA     CNTALY                                                                   1762          0175
 0120                CMPA    JJ                                                                       1764          0176
 0121                TPL     PACK2                                                                                  0177
 0122                STQ     0,4                                                                      1770          0178
 0123                AOS     CNTALY                                                                   1800          0179
 0124                ADX4    1,DU                                                                     1810          0180
 0125        PACK2   LDA     II                                                                                     0181
 0126                STZ     VECTOR+1,AL                                                                            0182
 0127        PACK1   AOS     II                                                                       1820          0183
 0128                LDA     II                                                                       1822          0184
 0129                CMPA    KK                                                                       1830          0185
 0130                TMI     PACK0                                                                    1840          0186
 0131                JOVIAL                                                                           1850          0187
 0132                IF CNTALY EQ 0 $                                                                 1860          0188
 0133                        BEGIN                                                                    1870          0189
 0134                        IO103(3,PRNT,0,WARN1,4) $                                                1880          0190
 0135                        PDUMP(DOCNUM,DOCNUM,1) $                                                 1890          0191
 0136                        GOTO SETDOC $                                                            1900          0192
 0137                        END                                                                      1902          0193
 0138        BLDOUT.                                                                                  1905          0194
 0138                DIRECT                                                                           1910          0195
 0139                LDA     DOCNUM                                                                   1920          0196
 0140                ALS     18                                                                       1930          0197
 0141                ADA     CNTALY                                                                   1940          0198
 0142                STA     OUTVEC+1                                                                 1950          0199
```

Figure V.29a. (Continued)

```
12736 01  04-29-71   18.671  *JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS
JOVIAL VERSION -- NUMBER 46
 0143         JOVIAL                                                    1960     0200
 0144         IO1C4(4.COMVEC,0,OUTVEC($1$),CWTALY+1) $                  1970     0201
 0145         DIRECT                                                             0202
 0146         LDA      DOCNUM                                                    0203
 0147         STA      OUTVEC+1                                                  0204
 0148         JOVIAL                                                             0205
 0149         FOR I=0,1,CWTALY-1 $                                               0206
 0150                  BEGIN                                            2020     0207
 0151                  DIRECT                                           2030     0208
 0152         LDA      I                                                2040     0209
 0153         LDQ      OUTVEC+2,AL                                               0210
 0154         QRL      18                                               2060     0211
 0155         STQ      OUTVEC                                           2070     0212
 0156                  JOVIAL                                           2080     0213
 0157                  IO105(4,PNTVEC,0,OUTVEC($0$),2) $                         0214
 0158                  END                                              2100     0215
 0159     ZEROIT.                                                       2110     0216
 0159           FOR I=0,1,CWTALY+1 $                                    2120     0217
 0160                  BEGIN                                            2130     0218
 0161                  DIRECT                                           2135     0219
 0162         LDA      I                                                2140     0220
 0163         STZ      OUTVEC,AL                                        2150     0221
 0164                  JOVIAL                                           2160     0222
 0165                  END                                              2170     0223
 0166     SETDOC.                                                       2180     0224
 0166           DOCNUM=INWRD($2$) $                                     2190     0225
 0167           GOTO SRCHO $                                            2200     0226
 0168           TERM PROCESS $                                          2210     0227
```

Figure V.29a. (Concluded)

MAXCOR - the number of words of storage required by the complete file of dictionary stem concept cluster

CLUSIZ - the number of Concept/Weight pairs per Cluster

CLUMAG - the number of words of storage per Cluster (normally = CLUSIZ+2)

CWMAXM - maximum number of Concept/Weight pairs allowed per compressed Document Concept Vector

DCVSIZ - the number of words of storage per DCV (normally = CWMAXM+2)

2) Assign Tapes:

I/O File	File Code	Description
CWPAIR	10	Input of Stem Concept Clusters
COMVEC	11	Output of DCV
PNT VEC	12	Output of Concept/Document Pairs
MICRON	13	Input of Microconcordances

V.7.5 Production Run Summary - DCVGEN

Processor Time	1.3722	CPU hours
Elapsed Time	5.3	hours

Approximately 17.5×10^6 microconcordances were input representing 36,254 documents. Thirty-five null vectors were generated (i.e., 35 abstracts contained no dictionary stems).

V.7.6 Introduction - Routine BLD.STR

Routine BLD.STR generates the inverted list of document accession numbers used to index the document concept vectors by concept number. Implementation of the inverted list structure for the document concept vector file reduces the retrieval of these vectors to a highly efficient, two-level search, further enhanced by the flexibility of random disc access.

As a concept is identified in a query vector during the Query Analysis phase of the retrieval routine, RETRIV, all document accession numbers related to that concept are read from random disc file. It is this file, generated by BLD.STR, which enables the system to construct a record of distinct document accession numbers for only those documents which correlate with the input query. Query Document correlation is performed on only those DCV's identified previously during query-analysis.

V.7.7 Flowchart Comments - Routine BLD.STR Refers to Figure V.30

1) Open all I/O files
2) Set stem LSTSIZ, the counter of document accession numbers, to 0. Preset CONNUM, the current concept number, to 1. Store CONNUM in the first entry of the document accession number buffer, array DOCLST.
3) Read a Concept/Document pair from input file CONDOC into array pair.
4) CONDOC EQ EOF?

   Yes ⇒10 No ⇒5
5) Is the concept number of the current Concept/Document pair equal to CONNUM?

   Yes ⇒8 No ⇒6
6) Is the concept number of the current Concept/Document pair equal to CONNUM+1?

   Yes ⇒10 No ⇒7
7) Inform the user that a concept number has somehow been deleted, close all I/O files, and terminate the job.
8) Is LSTSIZ, the number of document accession numbers entered in array DOCLST, greater than or equal to the maximum number of entries allowed, preset by Define "LSTMAX"?

   Yes ⇒3 No ⇒9
9) Add the document accession number of the current Concept/Document pair to array DOCLST. Increment LSTSIZ and C(X5), the index to array DOCLST.
10) Write the contents of DOCLST in output file STRING.
11) CONDOC EQ V(EOF)?

   Yes ⇒12 No ⇒13

12) Close all I/O files and terminate the job.

13) Zero-fill array DOCLST.

14) 0 = LSTSIZ; reset item CONNUM with the concept number of the current concept/document pair.

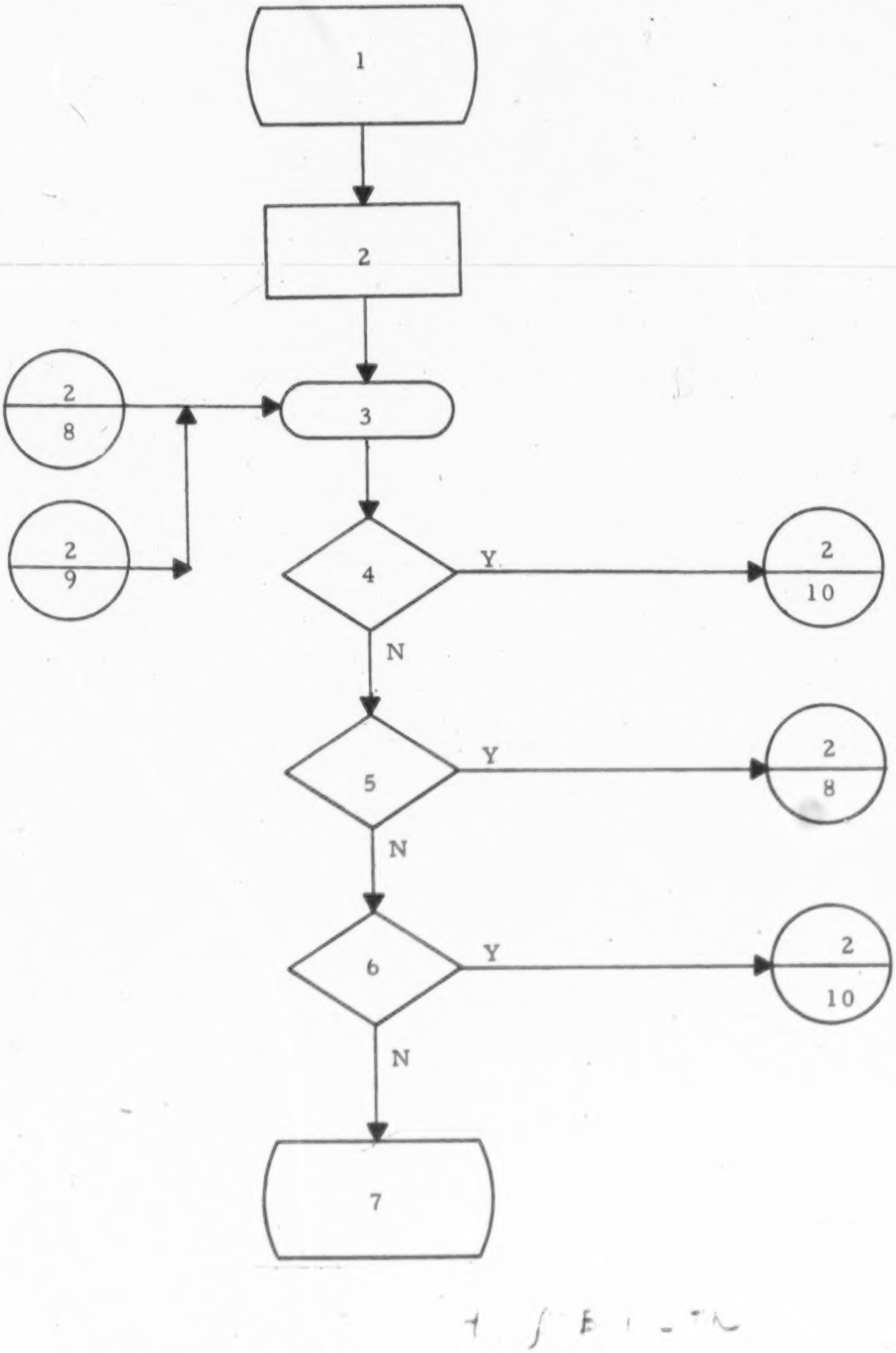

1
2
3
2
8
2
9
4
Y
2
10
N
5
Y
2
8
N
6
Y
2
10
N
7

Figure V. 30. Flowchart of BLD. STR

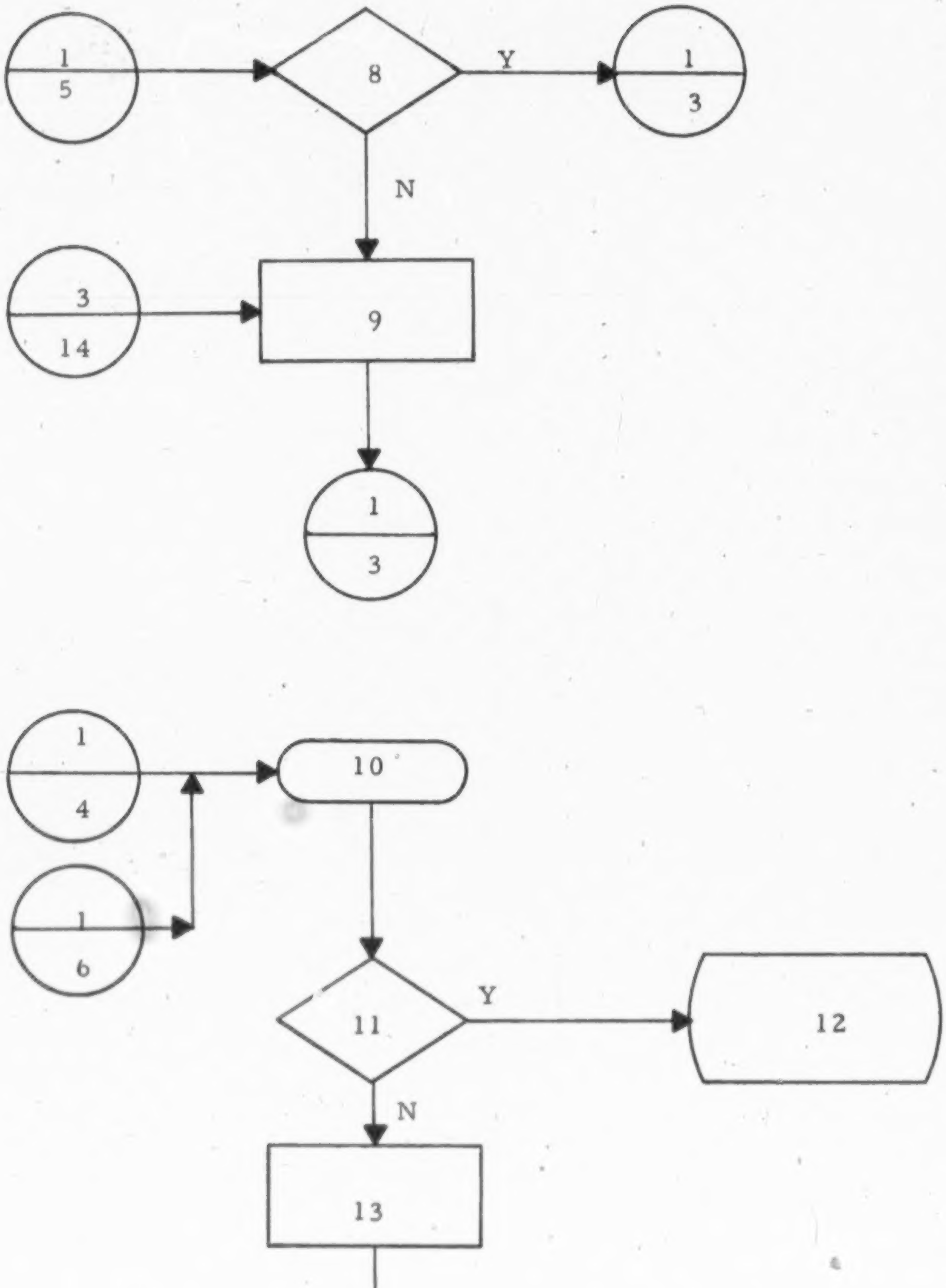

1
5
8
Y
1
3
N
3
14
9
1
3
1
4
10
1
6
11
Y
12
N
13
3
14

Figure V. 30. (Continued)

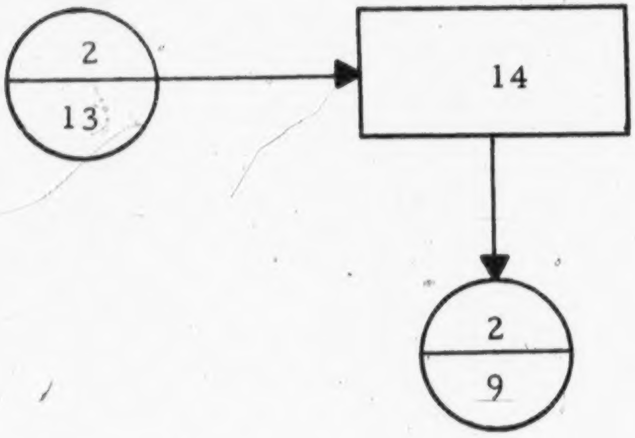


Figure V. 30. (Concluded)

V.8 SUMMARY OF THE BATCH SYSTEM

The Batch System as it has been described in the preceeding pages is a complete document analysis tool, although it is by no means a finished production system for general use. In order to decrease machine time, significant portions of the code have been made data dependent to a large degree. In addition, the statistical measures used to separate content from non-content stems during the analysis phase are subject to further testing. The co-occurrence stem clustering routine produces usable output in a fraction of the machine time estimated for some algorithms which have been suggested to solve the same problem. Again, detailed inspection of the semantic domains identified by this process needs to be done, in order to verify that the members of a concept are indeed semantically related.

The following table is a detailed breakdown of the Batch System production, showing tape reel numbers (R.A.D.C. tape serial numbers), tape contents, et cetera.

Table V.2 Explanation

Series II - 1967 Psychological Abstracts
Series III - 1968 Psychological Abstracts
Series IV - 1969 Psychological Abstracts

The tables contain a complete list of tapes, their contents, and the RADC serial numbers, including dates processed and routines used for the processing.

Routine Name	TAPES		Description of Contents
	Input	Output	
CONVRT Date: 11/20	42808		Abstract data base with bibliographic data in IBM 8-Bit character set
		44361	Files 1 and 2 of 42808 converted to HISI 635 character set
		45220	File 3 of 42808 converted to HISI 635 character set
GURGLE Date: 11/20	44361		Processed First
	45220		Processed Second
		45143	Isolated Abstracts
		45142	Isolated Bibliographic Data
MIC.CON Date: 12/01	45143		Isolated Abstracts
ABSTRACT # 1-13825		45228	Concordances: Stem-D/W Pair in alpha-collating sequence per document
		43804	Leaders: Total count of all stems per document
SRTMRG Date: 12/3/70	45228		Microconcordances in alpha-collating sequence
CONVRT Date: 11/20	44512		Abstract data base with bibliographic data in IBM 8-bit character set
		43553	Files 1 and 2 of 44512 converted to HISI 635 character set
		45219	File 3 of 44512 converted to HISI 635 character set
GURGLE Date: 11/25	43553		Processed First
	45219		Processed Second
		45210	Isolated Abstracts
		45214	Isolated Bibliographic Data

Table V. 2. Tape Charts

Routine Name	TAPES		Description of Contents
	Input	Output	
MIC.CON Date: 12/2/70	45210		Isolated Abstracts
ABSTRACTS # 138-26-22493		43828	Concordances: Stem-D/W pair in alpha-collating sequence per document
		44897	Leaders: Total count of all stems per document
SRTMRG Date: 12/07/70	43828		Microconcordances
		45098	Microconcordances in Alpha-collating sequence
CONVRT Date: 11/20	43437		Abstract data base with bibliographic data in IBM 8-bit characters
		43633	Files 1 and 2 of 43437 converted to HISI 635 character set
		45200	File 3 of 43437 converted to HISI 635 character set
GURGLE Date: 11/25	44713		Processed First
	45200		Processed Second
		45213	Isolated Abstracts
		45212	Isolated Bibliographic Data
MIC.CON Date: 12/4/70	45213		Isolated Abstracts
ABSTRACTS # 22494-36254		45094	Concordances: Stem-D/W pairs in alpha-collating sequence per document
		42991	Leaders: Total count of all stems per document
SRTMRG Date: 12/14/70	45094		Microconcordances

Table V.2. (Continued)

Routine Name	TAPES		Description of Contents
	Input	Output	
		43405	Microconcordances in alpha-order reel # 1 of 2 reel file
		45276	Reel # 2 of 2 reel file
LEADERS Date: 1/23	43804		Leaders, Series II, Document # 1-13825
	44897		Leaders, Series III, Document # 13826-22493
	42991		Leaders, Series IV, Document # 22494-36254
		43925	Merged leaders in related document order
MERGER Date: 2/2/71	44119		Alpha-ordered concordances Series II
	45098		Alpha-ordered concordances Series III
	43405		Alpha-ordered concordances Series IV 1 of 2
	45276		2 of 2
		44609	Merged Alpha-ordered concordances 1/3
		44601	2/3
		44608	3/3
SF1.B Date: 2/4/71	44609		Merged Alpha-ordered microconcordances 1/3
	44601		2/3
	44608		3/3
	43925		Merged Document Leaders
	Run	43960	Stem-V_c (Alpha-ordered)
	#1	43346	Reduced Master Concordances (D/W pairs by descending weight)

Table V.2. (Continued)

Routine Name	TAPES		Description of Contents
	Input	Output	
	Run	45216	Stem-V_c-σ_c-σ_c/V_c (Alpha-ordered)
	# 2	43961	Reduced Master Concordances (D/W pairs by descending weight)
SORT Date: 2/22/71	45216		Stem-V_c-σ_c-σ_c/V_c (Alpha-ordered)
		05443	Stem-V_c-σ_c-σ_c/V_c (Entries with σ_c.LQ.2 deleted)
SPLIT Date: 2/24/71	05443		Stem-V_c-σ_c-σ_c/V_c (With $\sigma_c > 2$ only)
		43337	Stem-V_c (Alpha-order of stems)
SF2.B Date: 2/25/71	43961		Reduced Master Concordances (D/W pairs by descending weight)
	43337		5,000 alpha-ordered stems with highest V_c
		45226	Master concordances of 5,000 stems with highest V_c (D/W pairs by ascending document number)
CLUSTR.0 Date: 3/20/71	45226		Master concordances of 5,000 stems with highest V_c (D/W pairs by ascending document number)
		45224	All cos $\phi > 0.0$
SYSTEM SORT	45224		All cos $\phi > 0.0$
		45206	All cos ϕ's by descending cos ϕ
CLUSTR.1 Date: 3/22/71	45206		All cos ϕ's by descending cosϕ
		45089	1,500 distinct stems (concept centers) (by ascending concept number)
CLUSTR.2 Date: 3/22/71	45089		1,500 distinct stems (concept centers) (by ascending concept number)

Table V.2. (Continued)

Routine Name	TAPES		Description of Contents
	Input	Output	
	45226		Master concordances of 5,000 stems with highest V_c (D/W pairs or ascending document no.)
		45209	CLUSTERS: Each dictionary stem with 4 or fewer C/W pairs (C/W pairs ordered by descending weight [F.P.])
EXTCON Date: 3/25/71	45209		All clusters (alpha-ordered)
		45327	All clusters + w/concept 1501 for all null vectors (alpha-ordered)
SYSTEM MERGE	45228		Microconcordances: Document # 1-13825
Date: 3/25/71	43828		Microconcordances: Document # 13826-22493
	45094		Microconcordances: Document # 22494-36254
		43595	Microconcordances merged by ascending document # 1 of 3
		44589	Microconcordances merged by ascending document # 2 of 3
		44108	Microconcordances merged by ascending document # 3 of 3
DCVGEN Date: 3/29/71	43595		Microconcordances merged by ascending document # 1 of 3
	44589		Microconcordances merged by ascending document # 2 of 3
	44108		Microconcordances merged by ascending document # 3 of 3
	45327		All clusters (null clusters contain miscellaneous concept #1501)
		42592 43591	Backup Copy Document Concept Vectors

Table V.2. (Continued)

Routine Name	TAPES		Description of Contents
	Input	Output	
		43491	Concept/Document Pairs 1 of 2
		43599	Concept/Document Pairs 2 of 2
SYSTEM SORT	43491		Concept/Document Pairs 1 of 2
Date: 3/31/71	43599		Concept/Document Pairs 2 of 2
		44156	Concept/Document Pairs ((1) Ascending con. #; (2) Ascending doc. # 1 of 2
		45351	Concept/Document Pairs ((1) Ascending con. #; (2) Ascending doc. #) 2 of 2
BLD.STR Date: 4/2/71	44156	-	Concept/Document Pairs ((1) Ascending con. #; (2) Ascending document #) 1 of 2
	45351		Concept/Document Pairs ((1) Ascending con. #; (2) Ascending document #) 2 of 2
		45352	Accession numbers 1-2000 blocked by concept number

Table V.2. (Concluded)

SECTION 6

SECTION VI

The On-Line System: Implementation Details

This section deals with the implementation of the on-line system on a more detailed level than does Section IV. First the general philosophy of on-line systems is developed, with specific reference to the present retrieval program. Then a functional description of each portion of the processor is presented in some detail. Covered in this portion are the following topics: The system executive monitor, the user functions, program and procedure overlays, other procedures and functions called by both the executive and the overlays. The next subsection deals with the logical structure of the system, both as it resides on permanent storage and also as it is loaded into the computer. Included in this subsection are the following topics: a discussion of major problems encountered in the assembly of the completed system, the use of independently compiled routines, and sample core maps, illustrating the system as it resides in core storage during execution of the on-line retrieval process.

The next subsection deals with actually putting the system into operation, and includes a discussion of the program deck setups necessary to operate the system, as well as a discussion of methods of preserving the generated overlays and core-images. Following this subsection, there is a general discussion of the various external files needed during the operation of the system, including size estimates for various configurations in which the retrieval system can run.

Following the discussion on files used by the system is a detailed discussion of the executive and how it functions in the overall design, including tables, variables, listings, and flowcharts with extensive commentary. The section concludes with a discussion of the future direction of the on-line system, with particular emphasis on further segmentation, additional functions, and methods of speeding up telecommunications with the user.

VI. 1 INTRODUCTION TO THE IMPLEMENTATION OF THE ON-LINE SYSTEM: GENERAL DESIGN PHILOSOPHY

The major underlying philosophy for this On-Line System concerns several related factors: ease of use by both beginners yet a sufficient depth to satisfy interested Information Retrieval specialists, economy of machine utilization, in terms of absolute quantities of computing resources used during execution, and reasonable execution speed in a real-time sense, so that users need not sit at an inactive terminal for extended periods of time, and a certain flexibility as to the extent of the complete system that any one user may want to utilize at any given retrieval session.

As in predecessor work (RADC-TR-70-44, contract F30602-69-C-0038), this Dialogue Processor caters to both the "tyro" user and the sophisticated experimenter, as transition from one category to the other is easily accomplished through the extensive tutorial nature of the man-machine dialogue. Thus, a beginner, through constant contact with the system gradually becomes experienced, and may bypass increasingly large portions of the tutorial contents as his expertise grows.

The overt manifestation of this dual nature is the existence of two forms of the man-machine dialogue messages: Terse and Verbose. It is the Verbose messages which contain most all of the tutorial materials, while the Terse forms attempt to convey much the same information without sending an undue amount of data to the user. In addition, once the user becomes curious about the other features of the system, he is gradually led into their use by appropriately placed cues. However, at any time the user may retreat into the relative security of the Verbose message forms by a simple request to the system.

The predecessor Dialogue Processor was not particularly economical to run, nor was its real-time execution speed adequate, both factors being beyond the control of the implementors. Given the environment of the Time-Sharing System and the programming language used, the system was a reasonable simulation of the present system. In terms of machine economy, the rigid storage layouts available in FORTRAN did not allow data-packing, while the nature of the machine-code generated by the TSS-FORTRAN compiler precludes economy of execution time and program storage*. This system required a core allocation of around 51K for the Time-Sharing System, although it had been run in slightly less space at times. All external files were sequential access, including the file of man-machine messages, thus eating up elapsed time in constant disc accesses. The present system solves most all of the problems listed above.

The present Processor has been coded entirely in JOVIAL, for reasons outlined in Section IV. 2. 5 of this report. The system runs as an ordinary slave batch program in the GECOS-III environment, with telecommunications handled through GEFRC (ultimately GERTS). The advantages of this method of operation are many. In the first place, unlike a Time-Sharing System program, a batch program does not compete with a host of other time-sharing users for the same core area, being subject to swap-out at each I/O request. In addition, I/O requests are handled directly by the GECOS-III modules needed, instead of indirectly through the TSS executive. Also, access is permitted to batch-world files not resident in the permanent file system, a necessary restriction of TSS programs and subsystems not running as priviledged slaves.

Through the use of the $ USE .RTYP control card and the file cards $ DAC 05 and $ DAC 06, the Dialogue Processor is connected to the remote user at execution time. Thereafter, the running program treats the user teletypewriter as any ordinary GECOS-III file.

---

* TSS FORTRAN generates "self-relocatable" code, using the instruction-counter as the local "base-address-register," thus requiring additional numbers of instructions to perform operations which could be carried more simply in ordinary relocatable code. In addition, because of the nature of the code, linkages to subroutines are of necessity quite complex and time-consuming to run.

The advantages are many. If desired, the system could be run as a batch retrieval system, simply by removing the $ DAC cards and the $ USE .RTYP GELOAD directive. In this case a deck of cards would be provided to simulate the remote user's responses to the system dialogue.

Because the processor runs in the batch world and not Time-Sharing, response to teletypewriter input and output is quite fast, even when the computing system is quite heavily loaded. The only note of caution is that, unless the machine operator manually raises the urgency of the Dialogue Processor after it is in execution, the job is subject to swap-out for varying periods of time, depending upon the computing load at the moment. The manifestation of this annoying condition is a completely "dead" remote terminal, either after the program has sent a message to the user, or after the user has typed something in reply to a question or has entered textual material for analysis by the system. The direct remedy is a quick telephone call to the operator, explaining that the user's batch job is in reality on-line and should not be allowed to be swapped out.

Besides the markedly decreased response times of the present processor, core storage requirements have been drastically reduced in comparison to the previous system. Only a 29K core allocation is necessary in order to execute the system, including the main segment, file control blocks and all overlaid portions of the processor. Of course, further segmentation is possible, and this matter is discussed in Section VI. 6. 1 of this report. However, further work in this particular area is not envisioned in the present effort, although it should be seriously considered in future work because of increased computing loads and more stringent limits on the sizes of executing programs.

The net result of the economies mentioned above is that the system is allocated for execution very quickly, the user obtains a minimum delay in his communication with the system, and machine usage in terms of CPU-seconds is quite light. For further details on run summaries see Section VI. 5. 4.

VI. 2 FUNCTIONAL DESCRIPTION OF THE ON-LINE SYSTEM

In this subsection the individual modules which go together to make up the entire On-Line System are discussed in detail, their functions outlined, and flowcharts provided to clarify the logic.

VI. 2. 1 The Executive Monitor

As in any large and complex system there must be one portion which performs the supervisory function, so it is in the Dialogue Processor. Included in what is termed the "monitor" are all the core resient portions of the system, encompassing the control over program flow, input/output interface with the on-line user, utility routines for table searching, as well as routines for conversion from binary integer to BCD representation and vice-versa.

VI. 2. 1. 1 Supervisory Routines.

Classed as supervisory routines are DIALOG, CHAT, and OPT. Each is discussed in turn below.

DIALOG: This is the main program for the Dialogue Processor and serves as the driver for the rest of the On-Line System. Dialog performs system initialization and loads the overlays as they are needed by the different functional parts of the core resident system. The logic of DIALOG is illustrated in Figure VI. 1.

CHAT: The CHAT portion of the resident carries on the conversation with the user about documents retrieved during the current retrieval cycle. Most generally, on the experienced user calls upon the system functions directly through OPT, the next routine to be discussed. Figure VI. 2 is the flowchart for CHAT.

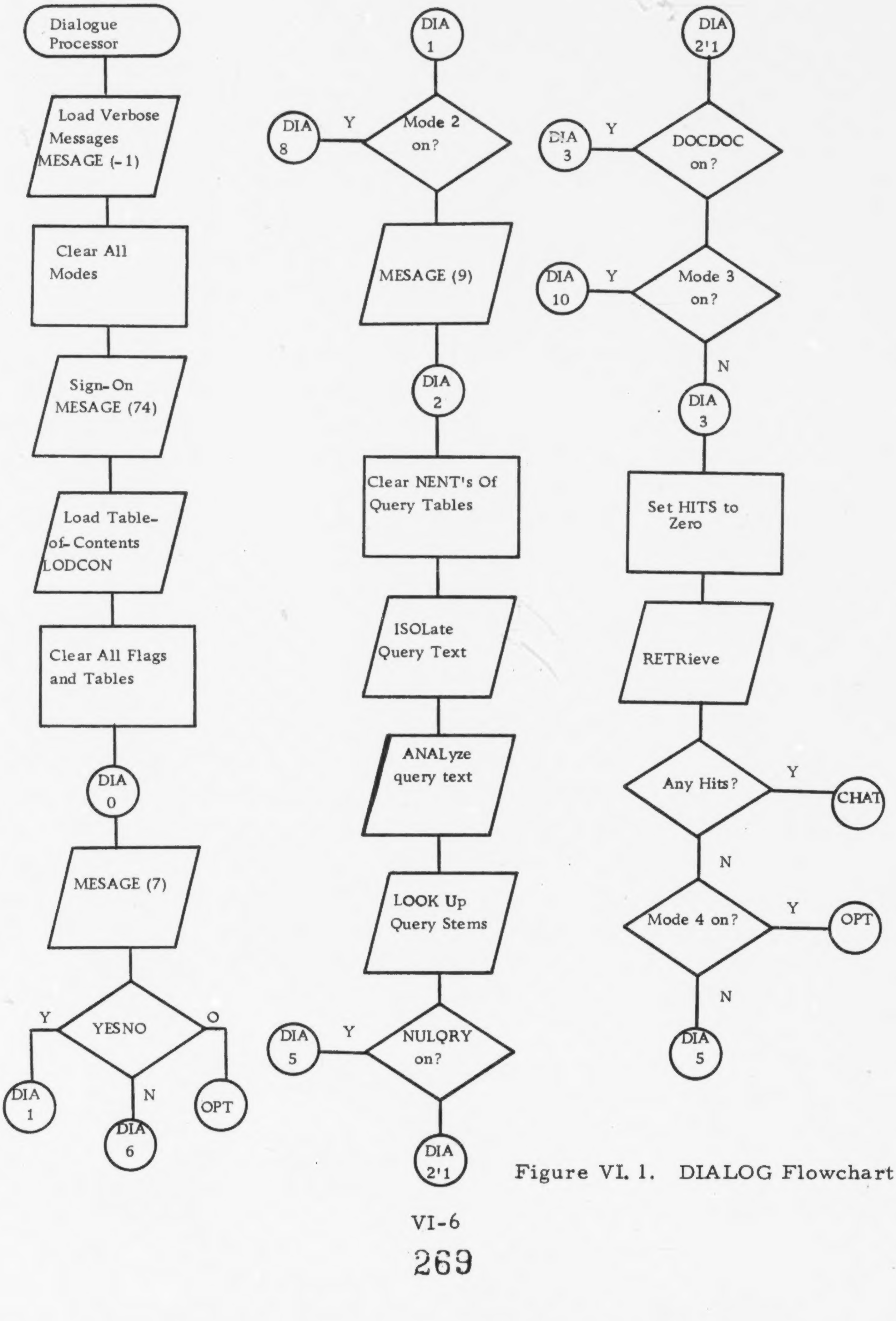

Dialogue Processor
Load Verbose Messages MESAGE (-1)
Clear All Modes
Sign-On MESAGE (74)
Load Table-of-Contents LODCON
Clear All Flags and Tables
DIA 0
MESAGE (7)
YESNO
Y
N
O
DIA 1
DIA 6
OPT
DIA 1
Mode 2 on?
Y
DIA 8
MESAGE (9)
DIA 2
Clear NENT's Of Query Tables
ISOLate Query Text
ANALyze query text
LOOK Up Query Stems
NULQRY on?
Y
DIA 5
DIA 2'1
DIA 2'1
DOCDOC on?
Y
DIA 3
Mode 3 on?
Y
DIA 10
N
DIA 3
Set HITS to Zero
RETRieve
Any Hits?
Y
CHAT
N
Mode 4 on?
Y
OPT
N
DIA 5

Figure VI. 1. DIALOG Flowchart

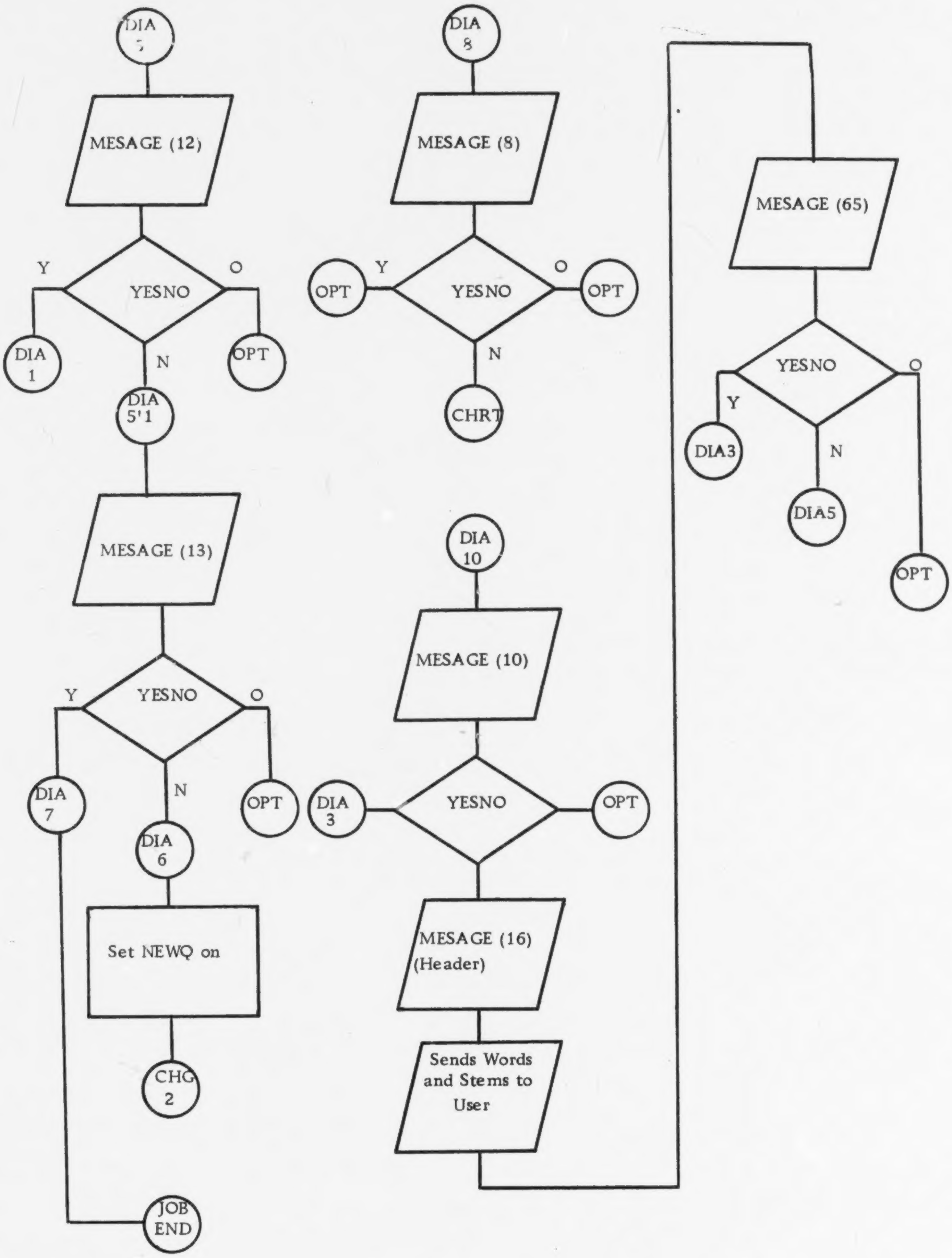

DIA 5
MESAGE (12)
Y
YESNO
O
DIA 1
N
OPT
DIA 5'1
MESAGE (13)
Y
YESNO
O
DIA 7
N
OPT
DIA 6
Set NEWQ on
CHG 2
JOB END
DIA 8
MESAGE (8)
OPT
Y
YESNO
O
OPT
N
CHRT
DIA 10
MESAGE (10)
DIA 3
YESNO
OPT
MESAGE (16) (Header)
Sends Words and Stems to User
MESAGE (65)
YESNO
O
Y
DIA3
N
DIA5
OPT

Figure VI. 1. (Concluded)

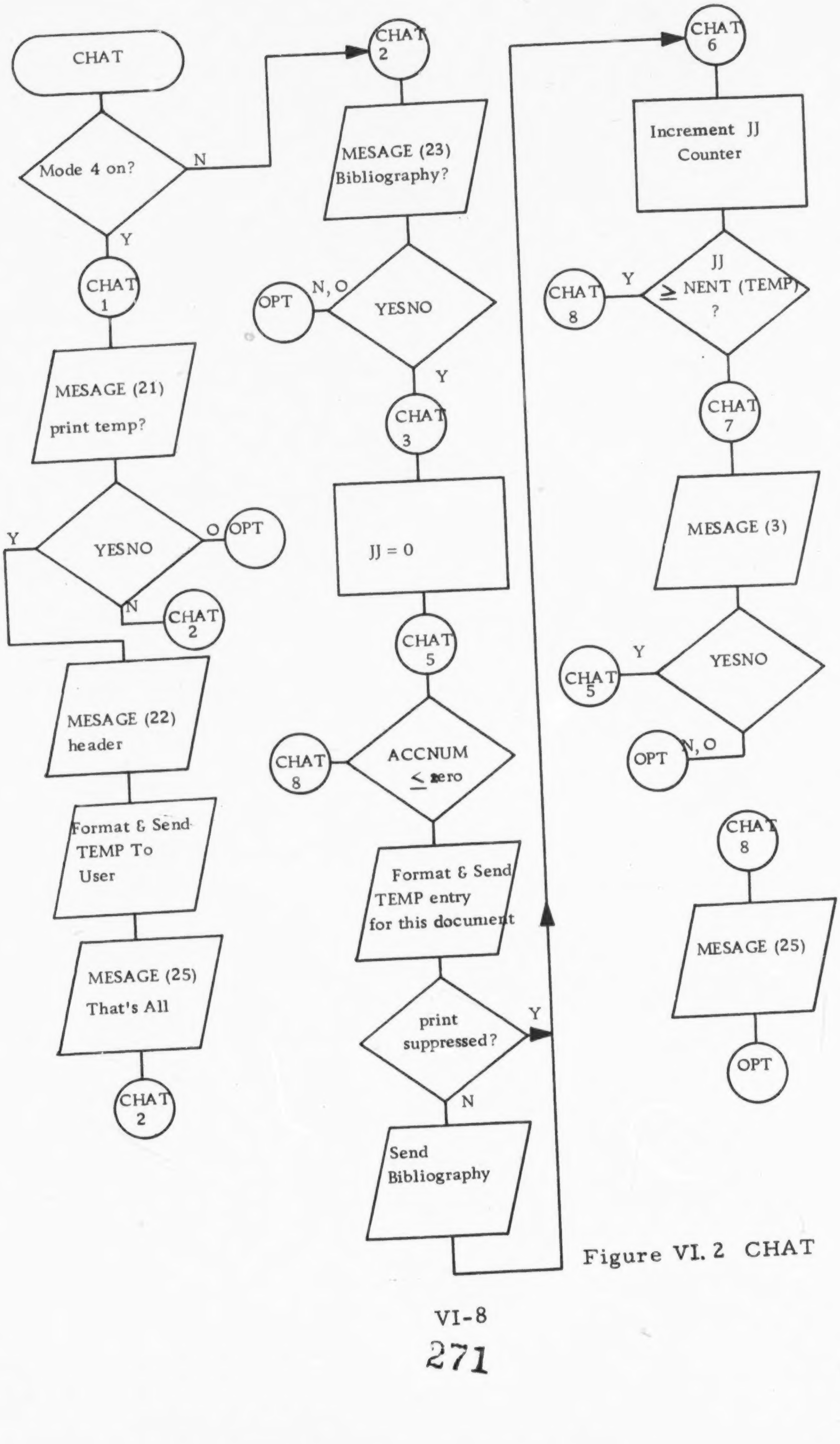

CHAT
Mode 4 on?
N
Y
CHAT 1
MESAGE (21) print temp?
YESNO
Y
O
OPT
N
CHAT 2
MESAGE (22) header
Format & Send TEMP To User
MESAGE (25) That's All
CHAT 2
CHAT 2
MESAGE (23) Bibliography?
YESNO
N, O
OPT
Y
CHAT 3
JJ = 0
CHAT 5
ACCNUM ≤ zero
CHAT 8
Format & Send TEMP entry for this document
print suppressed?
Y
N
Send Bibliography
CHAT 6
Increment JJ Counter
JJ ≥ NENT (TEMP) ?
Y
CHAT 8
CHAT 7
MESAGE (3)
YESNO
Y
CHAT 5
N, O
OPT
CHAT 8
MESAGE (25)
OPT

Figure VI. 2 CHAT

OPT: The main driving portion of the Dialogue Processor is the function OPT. Through this routine the user, whether experienced or beginning, obtains access to the various sections of the on-line system which best suit his purposes. OPT serves also as a major tutorial subsystem, guiding the beginning user in the use of the available functions and system modes. At any point in the man-machine dialogue, when the user is asked to respond to a "yes-no" question, the answer OPTIONS or OPT is also recognized, at which point the sequence of operations is interrupted and the OPT function takes over.

Unless OPT is entered directly from YESNO (see Appendix B.3.4, and the detailed discussion of YESNO below), an option name is requested from the user. The user further has the tutorial aid of the pseudo-option HELP. A cry for HELP causes an explanation of the available modes to be sent to the terminal user. Again, the available modes depend upon the setting of the program mode switch OPTION. If TERSE is "off," a rather detailed explanation of the modes and their usage is sent. Otherwise, a simple list is sent. Illegal or nonexistent replies are detected, and remedial action is taken by the processor in the form of a warning message and repetition of the request for a legal option.

Access to the various options is controlled by the use of table OPTNAM, with a corresponding JOVIAL switch OPTSW, which performs the selective branch into the called routine. Error checks are on two levels: illegal names are checked, and, as in the case of system error, a default "detent" on the OPTSW simply repeats the whole OPT sequence. Figure VI. 3 is the flowchart for OPT.

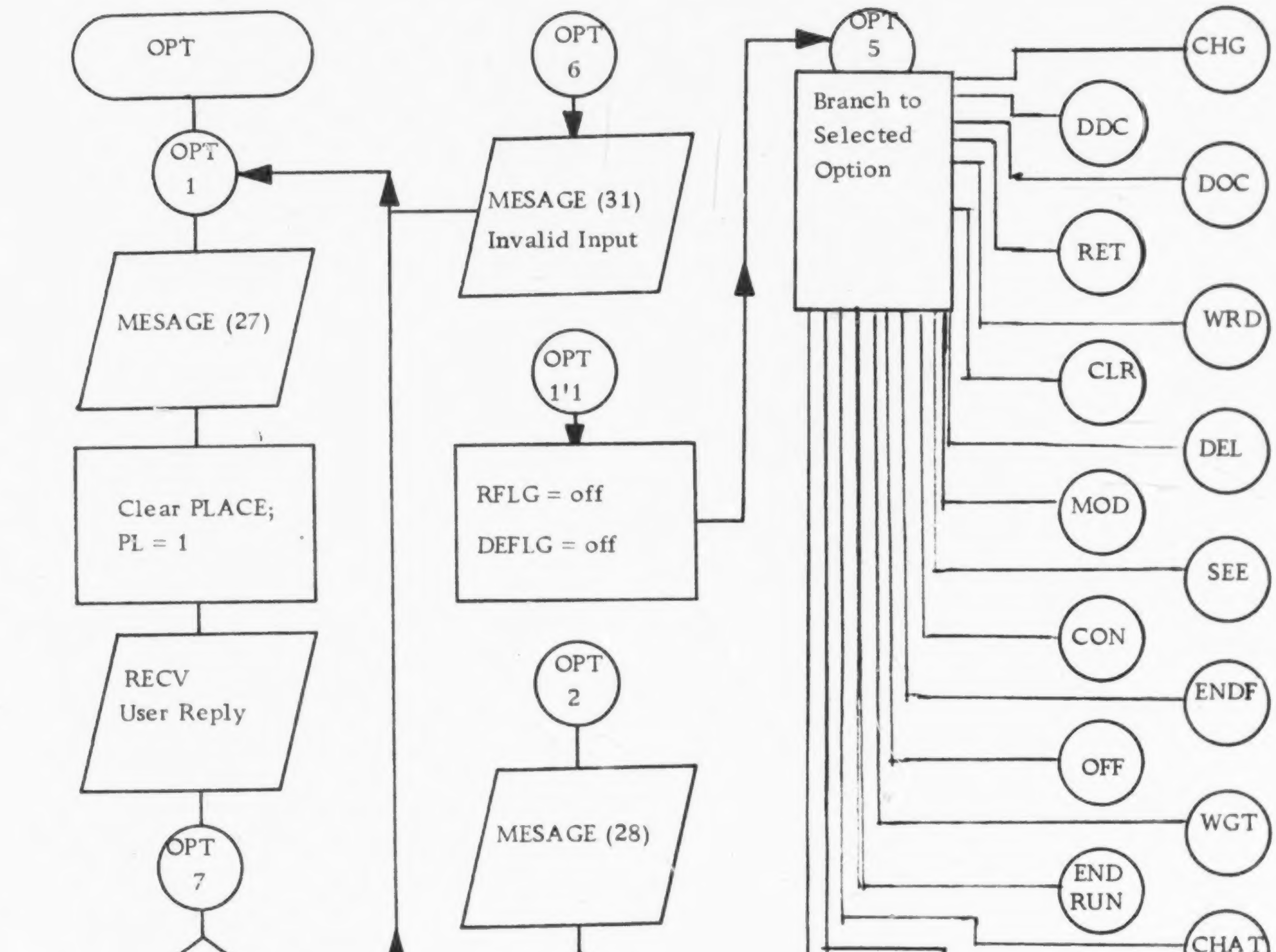

OPT
OPT 1
MESAGE (27)
Clear PLACE; PL = 1
RECV User Reply
OPT 7
Was it "HELP" ?
Y
OPT 2
N
Look up option in OPTNAM
A legal name?
Y
OPT 1'1
N
OPT 6
OPT 6
MESAGE (31) Invalid Input
OPT 1'1
RFLG = off
DEFLG = off
OPT 2
MESAGE (28)
Mode 5 on?
N
Y
OPT 3
MESAGE (29)
OPT 4
MESAGE (30)
YESNO
Y
N, O
OPT 5
Branch to Selected Option
CHG
DDC
DOC
RET
WRD
CLR
DEL
MOD
SEE
CON
ENDF
OFF
WGT
END RUN
CHAT
BLD
SKIM

Figure VI. 3. OPT Flowchart

VI. 2. 1. 2 Executive Utility Routines

Included in this portion of the resident system are independently-compiled and local (to the Dialogue Processor code) service routines, some called only by the resident system, others used both by the resident system and by the overlaid portions of the processor. Each is characterized by being executed from some portion of the executive and returning control to a point immediately after the call. The single exception to this condition is YESNO as employed by the sophisticated user.

MESAGE: (Independently-compiled). The Dialogue Processor "speaks" to the on-line user via a set of prestored messages, loaded from permanent files at system initialization. At appropriate points in the dialogue the procedure MESAGE is called, issuing the correct message from the table. MESAGE is also called at system initialization time to load the verbose messages initially before beginning dialogue execution.

Messages exist in two forms, as previously mentioned: Terse and Verbose. The mode flag TERSE controls which category is sent. TERSE may be toggled by the user through dialogue procedure CHG (Change-Modes function). For details on the operation of this function see Section VI. 2. 2 below. The prestored messages consist of one or more lines of BCD data. Since each message may have more than one line, some method is needed to keep track of the starting table address of each message, both for the terse and for the verbose types.

In addition, since the system may only be in one mode at a time, (either terse or verbose) with respect to message type, the two message tables are overlayed to conserve core space. Of course, when modes are switched, the opposite type of messages must be reloaded from external storage. However, this short delay is hardly noticed by the on-line user.

To implement the stored-message processor, two tables are preset when MESAGE is compiled: (also see Figure VI. 4).

SMX: Small message indices

LMX: Long message indices

Each table consists of two entries:

S/LMP: Pointers to message starting addresses

S/LML: Line count for each stored message

The message number is the explicit index into tables LMX and SMX, so that MESAGE obtains the output text information indirectly, using these pointer tables. Additional messages may be entered into the tables by presetting the appropriate entries in the pointer tables, and by adding the message text to the permfile containing the messages, assuring that sufficient space has been allocated in the message processor to hold the additional information.

The messages themselves are stored externally in two forms: ASCII (TSS standard format), complete with line numbers, and in BCD, suitable for direct loading into the program at execution time. The BCD forms are obtained from the ASCII files by executing the Time-Sharing command ASCBCD and answering the query LABELS? with the reply STRIP. As the loading program in the message processor has a cut-off of 72 characters per line, care must be taken to assure that no message has any more than this limit per line. Messages should terminate with the two characters "!1," in order to trim off trailing blanks during transmission to the user.

YESNO (local to the Dialogue Processor main-segment code):

At many points in the man-machine dialogue the user is required to answer questions "yes" or "no." Function procedure YESNO obtains the answer from the user and reports back to the calling routine. In addition, the user may elect to terminate whatever action the processor is currently undergoing by replying OPTIONS (or OPT) to the request for a yes-no response. In this case control passes from the question-issuing point to the dialogue procedure OPT.

SMX Table
SMP

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	20
21	22	24	25	26	27	28	29	30	31
32	33	34	35	36	37	39	40	41	42
43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	83	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

LMX Table
LMP

0	1	2	3	4	5	9	11	12	13
15	17	19	20	21	23	24	26	27	29
31	33	34	36	38	39	41	42	44	52
62	63	64	66	67	68	77	78	79	80
81	82	83	85	87	89	91	93	96	98
100	101	103	104	105	106	107	108	109	110
111	113	114	116	117	118	119	121	122	123
124	126	127	128	129	133	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

SML

0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	2	1
1	1	2	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	6	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

LML

0	1	1	1	1	4	2	1	1	2
2	2	1	1	2	1	2	1	2	2
2	1	2	2	1	2	1	2	8	0
1	1	2	1	1	9	1	1	1	1
1	1	2	2	2	2	2	3	2	2
1	2	1	1	1	1	1	1	1	1
2	1	2	1	1	1	2	1	1	1
2	1	1	1	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure VI. 4. Message Pointer Tables

As mentioned in the discussion of OPT above, another feature of function YESNO is the elimination of the need to type OPTIONS before typing the option desired, under certain conditions. This feature is open to the user once the mode flag OPTIONS (flag number 5) is set through the use of the CHG function. For the casual user as well as the sophisticated one, setting this option eliminates an additional step each time a new option is desired. However, there is a certain danger in using this shortcut, as a typing mistake will immediately return the user to OPT for a new choice of option names. It can never be the case, though, that a query may be lost this way or that any other vital information be lost to the system.

Unless mode flag 5 is on, YESNO returns one of three values to the calling routine:

0 (zero)	user replied "yes"
1	user replied "no"
2	user replied "options"

If none of the above replies was received by YESNO, and if the OPTION mode is on, then the user's reply is moved to the appropriate storage area directly in OPT itself. A special entry is provided in OPT to bypass a request for user input from the terminal. In the case of an error on the user's part, the standard error recovery process is followed in OPT: control cannot pass from OPT until a legal option name is entered at the remote terminal. Figure VI. 5 is the flowchart for YESNO.

SEND & RECV: In order to standardize telecommunication, all input and output to and from the remote user are channeled through RECV and SEND respectively. Besides the matter of standardization, there is the consideration of the future of the system. Should a decision be made at a later date to use a different type of remote access (such as direct calls to GERTS), then switch-over difficulties would definitely be minimized, as recoding would only be necessary for the terminal interface packages.

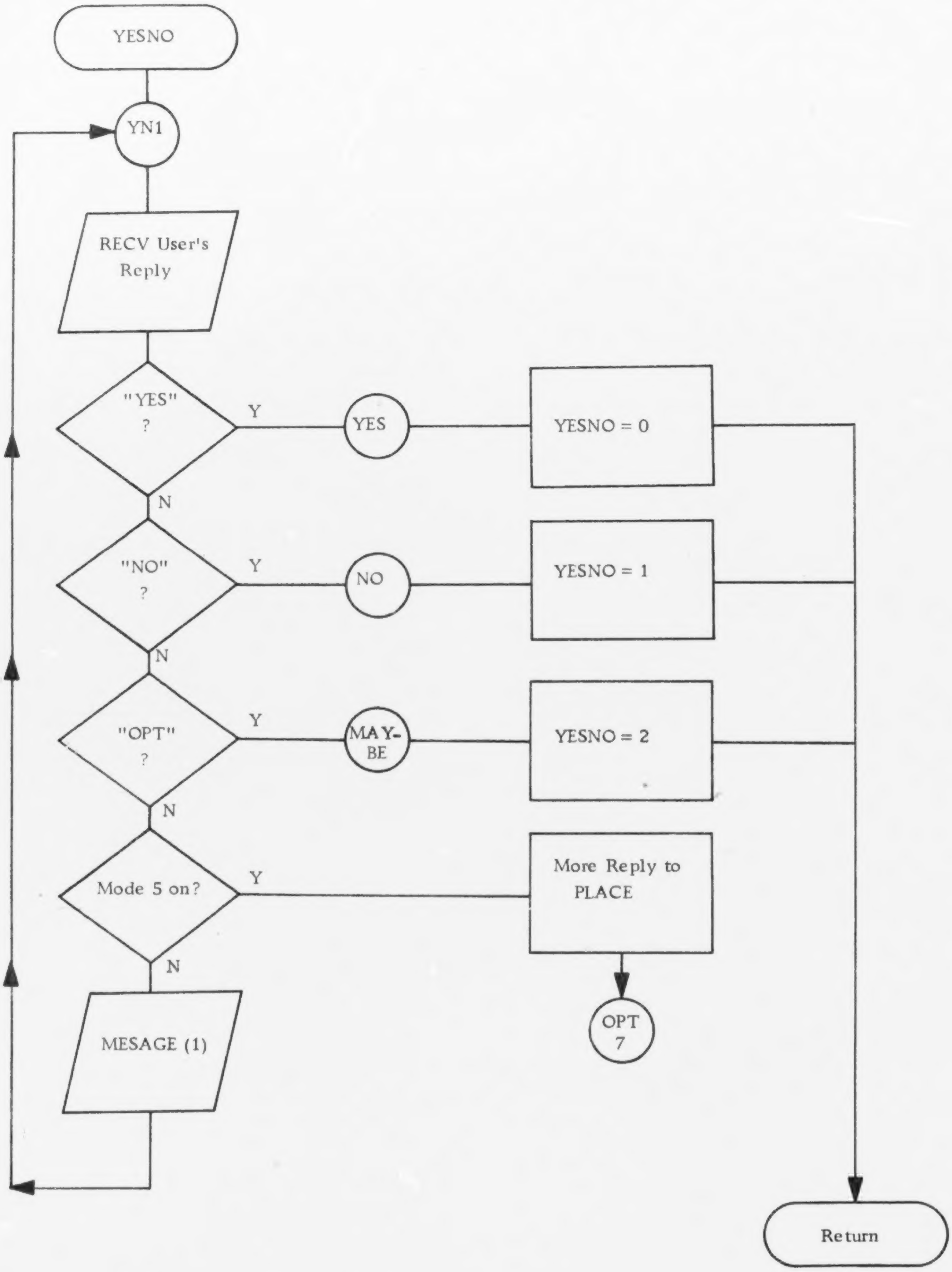

YESNO
YN1
RECV User's Reply
"YES" ?
Y
N
YES
YESNO = 0
"NO" ?
Y
N
NO
YESNO = 1
"OPT" ?
Y
N
MAY-BE
YESNO = 2
Mode 5 on?
Y
N
More Reply to PLACE
OPT 7
MESAGE (1)
Return

Figure VI. 5. YESNO Flowchart

As they are currently implemented, SEND and RECV write on file 06 and read from file 05 respectively. The I/O requests are not explicitely buffered in the JOVIAL code, but since these operations take place through a GEFRC interface, maximum utilization of the transmission period is taken by GECOS-III. The result of this feature is the following: even though the system may be resident for a considerable elapsed time, actual CPU usage (CPU-seconds) is quite light, considering the amount of actual machine code executed. SEND and RECV are compiled as separate procedures, in order to be accessible both from the main-segment and from the overlays.

BINBCD: (binary integer to BCD representation conversion) Numeric information to be transmitted to the terminal user must be converted to a character format before sending. BINBCD performs the conversion by the method of power division. The actual conversion takes place on the absolute magnitude of the number to be converted rather than treating negative numbers differently. The most significant six digits of positive numbers are converted, while only the most significant five digits of negative numbers may be converted, in order to leave room for the sign. BINBCD is written as a JOVIAL procedure with two parameters: 1) binary integer input, 2) BCD (character) output in one machine word.

BCDBIN: Information received from the terminal user must be converted into binary integer form before any computations may be performed on the data. Function procedure BCDBIN converts positive integers in character form to internal binary integers. Input parameter is the BCD number in one machine word; output is the assigned value of the function call.

HUNT1/HUNT2: These procedures perform binary (logarithmic) table searches on JOVIAL-organized internal tables either 1 or 2 machine-words in width. The technique is the standard one for such searches, and the logic for both routines is identical, except for table-address computation.

VI.2.2 User Functions

The On-Line System is compartmented into functional areas, coded as separate regions within the main code of the processor but not separately compiled. The name of each user function is the main entry point to the code area; secondary entries bear the initials of the function to which they belong. User functions may either be entered implicitly through the normal logic of the system, as is the case with the beginner, or explicitly, through a direct call (using OPT, as described above). The functions are described below in the order in which they appear in the control table in OPT. Also included is the text-line number of the start of the function code in the Dialogue Processor code.

CHG (4190): Change-modes function; flowchart is Figure VI. 6.

One of the controls the system user has over the operational logic is the ability to set any or all of five different flags. Flag-numbers range from one to five, with the value of zero interpreted as the end-of-information marker. Each flag is described below:

1. TERSE. Terse dialog. When this flag is referenced, its previous value is toggled. Consequently, TERSE may only be altered explicitly by the user and is not cleared at any time by the system. Use of the terse form is included to allow the elimination of much of the tutorial information contained in the user messages, with an eye to speeding up the man-machine dialogues for experienced users. If at any point the significance of a particular message is unclear, the user always has the option of returning to VERBOSE mode to clarify matters.
2. SKIP1. When flag 2 is set, initial query formation is skipped. As it should be recalled, initial query formation is done from text typed in by the system user. However, the sophisticated researcher may want to manipulate the

query concept vector directly, without interference from the automatic portions of the system. SKIP1 assures this category of user complete control over the contents of the query vector.

3. PRINTQ. Mode flag 3 permits an analysis of the query before a retrieval is performed. When it is on, mode three prints a table of query words and their corresponding stems, as produced by the analysis portions of the system. A direct examination of the query-concept-vector itself may be obtained through use of the WGT option, described below:

4. PRINTR. Mode flag 4 permits a detailed analysis of a retrieval, above and beyond the normal amount of information given about retrieved documents. Included in the information given under this option are the following items:
   - a) Accession number of retrieved document
   - b) Temporary I. D. assigned to this document by the on-line system
   - c) Its relative rank in the retrieval, the last time this particular document was retrieved
   - d) Whether or not its bibliographic information has already been printed
   - e) Whether or not this particular document was retrieved in the last retrieval cycle.
   - f) The correlation coefficient between the document-concept-vector for this document and the query-concept-vector at the time when it was last retrieved, (not necessarily by the last retrieval operation).

5. OPTION. Mode flag 5, when on, assumes the system is being used by a sophisticated user. The results of this setting are the following:

a) all options may be called explicitly by the user

b) user function OPT lists all the available functions instead of just ENDF, MOD, and DOC as for "tyro" users

c) any option name may be used in place of a reply to yes-no (as described in YES NO, above)

Mode changing numbers are comma-delimited and must be terminated with a pseudo-mode setting of "zero." No embedded blanks are allowed in the mode string; errors result in repetition of the question-answer sequence until a correct response is received by the system. Figure VI.6 illustrates the logic of CHG.

DDC (4540): Perform Document-document correlation.

The ability to substitute the document concept vector of any document or group of documents for the query concept vector is one of the most powerful features of the on-line system, allowing the user to retrieve documents which are similar in concepts and weights to previously retrieved documents (or any document of known accession number). User function DDC queries the user for the document numbers and builds a new query concept vector based on the document concept vectors for those accession numbers specified. After a legal query concept vector is built, a retrieval may take place normally, as if input text had been analyzed and the query vector built in the usual fashion. Figure VI.7 outlines the logic of Document-document correlation.

DOC (4960): Obtain information about retrieved documents.

Through this option the user may obtain printed copies of the abstracts retrieved during the last cycles, as well as any abstract of known accession number. Unless called explicitly by the user, DDC is entered normally by the system after any retrieval is performed. Figure VI.8 illustrates the logic of DOC.

RET (5950): Perform a retrieval, using the present query concept vector.

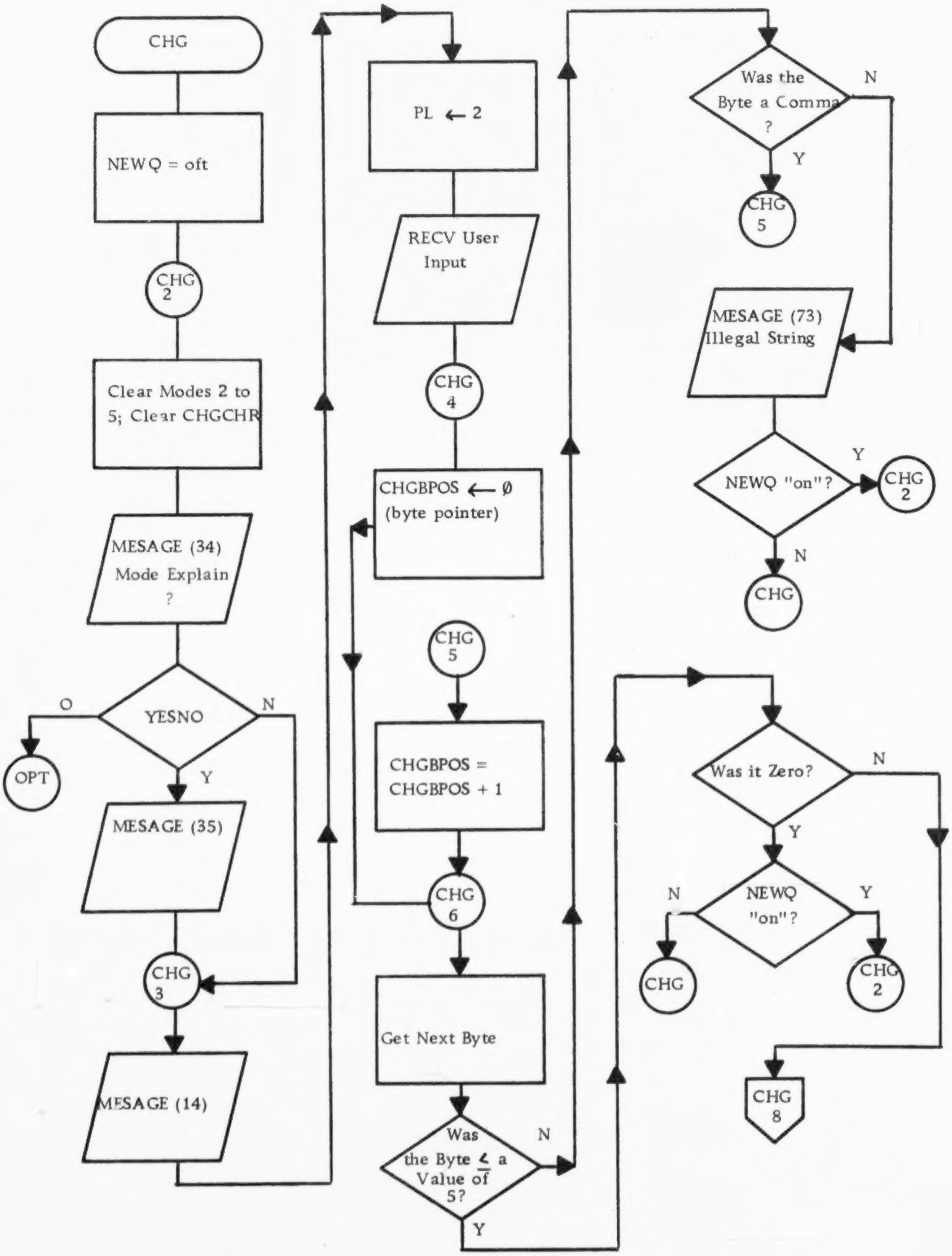

CHG
NEWQ = oft
CHG 2
Clear Modes 2 to 5; Clear CHGCHR
MESAGE (34) Mode Explain ?
YESNO
O
N
Y
OPT
MESAGE (35)
CHG 3
MESAGE (14)
PL ← 2
RECV User Input
CHG 4
CHGBPOS ← 0 (byte pointer)
CHG 5
CHGBPOS = CHGBPOS + 1
CHG 6
Get Next Byte
Was the Byte ≤ a Value of 5?
N
Y
Was the Byte a Comma ?
N
Y
CHG 5
MESAGE (73) Illegal String
NEWQ "on"?
Y
CHG 2
N
CHG
Was it Zero?
N
Y
NEWQ "on"?
N
Y
CHG
CHG 2
CHG 8

Figure VI. 6. CHG Flowchart

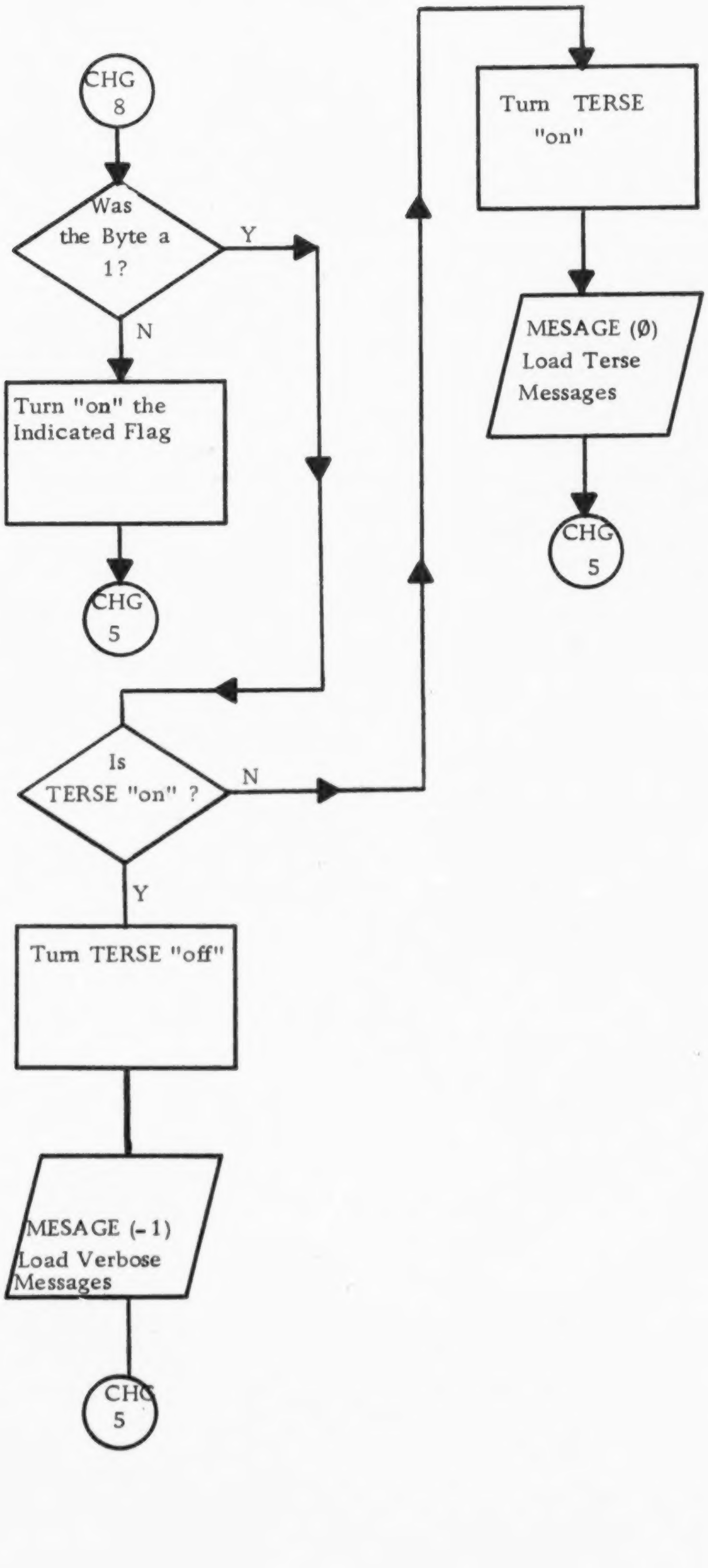

CHG 8
Was the Byte a 1?
Y
N
Turn "on" the Indicated Flag
CHG 5
Is TERSE "on" ?
N
Y
Turn TERSE "off"
MESAGE (-1) Load Verbose Messages
CHG 5
Turn TERSE "on"
MESAGE (0) Load Terse Messages
CHG 5

Figure VI. 6. (Concluded)

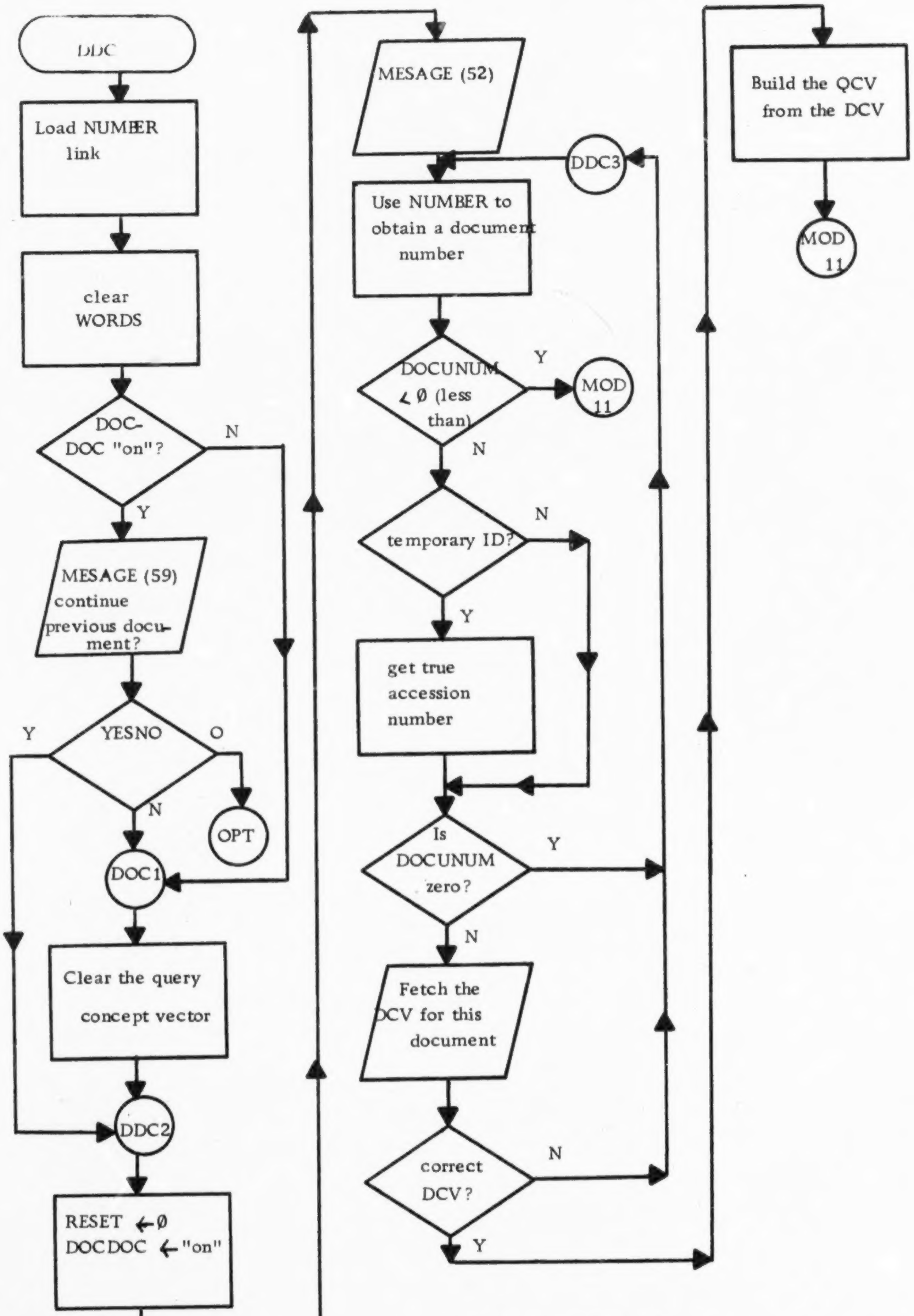

DDC
Load NUMBER link
clear WORDS
DOC-DOC "on"?
N
Y
MESAGE (59) continue previous document?
YESNO
Y
O
N
OPT
DOC1
Clear the query concept vector
DDC2
RESET ← 0
DOCDOC ← "on"
MESAGE (52)
DDC3
Use NUMBER to obtain a document number
DOCUNUM < 0 (less than)
Y
MOD 11
N
temporary ID?
N
Y
get true accession number
Is DOCUNUM zero?
Y
N
Fetch the DCV for this document
correct DCV?
N
Y
Build the QCV from the DCV
MOD 11

Figure VI. 7 DDC Flowchart

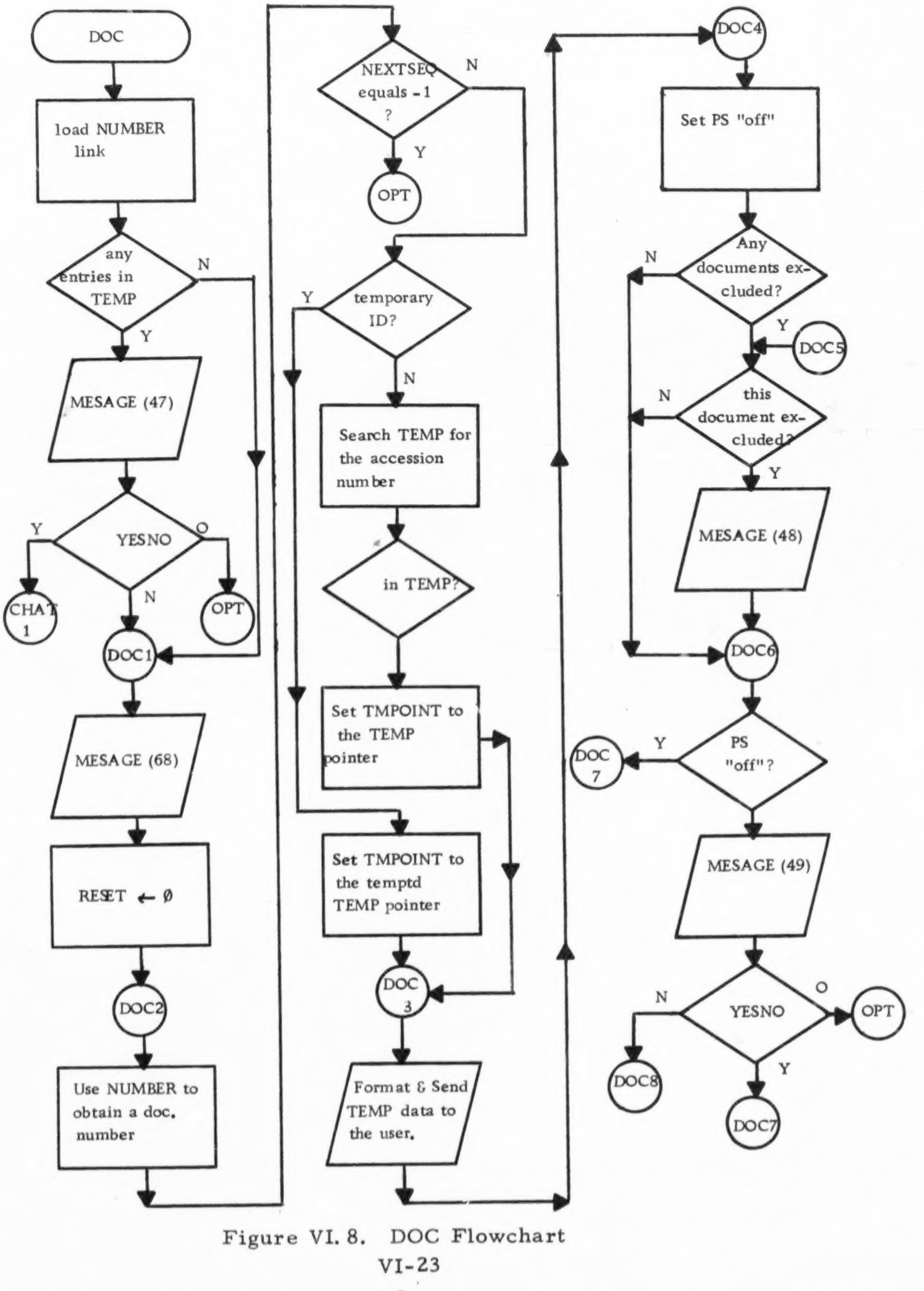

DOC
load NUMBER link
any entries in TEMP
N
Y
MESAGE (47)
YESNO
Y
O
N
CHAT 1
OPT
DOC1
MESAGE (68)
RESET ← 0
DOC2
Use NUMBER to obtain a doc. number
NEXTSEQ equals -1 ?
N
Y
OPT
temporary ID?
Y
N
Search TEMP for the accession number
in TEMP?
Set TMPOINT to the TEMP pointer
Set TMPOINT to the temptd TEMP pointer
DOC 3
Format & Send TEMP data to the user.
DOC4
Set PS "off"
Any documents excluded?
N
Y
DOC5
this document excluded?
N
Y
MESAGE (48)
DOC6
PS "off"?
Y
DOC 7
MESAGE (49)
YESNO
N
O
Y
OPT
DOC8
DOC7

Figure VI. 8. DOC Flowchart

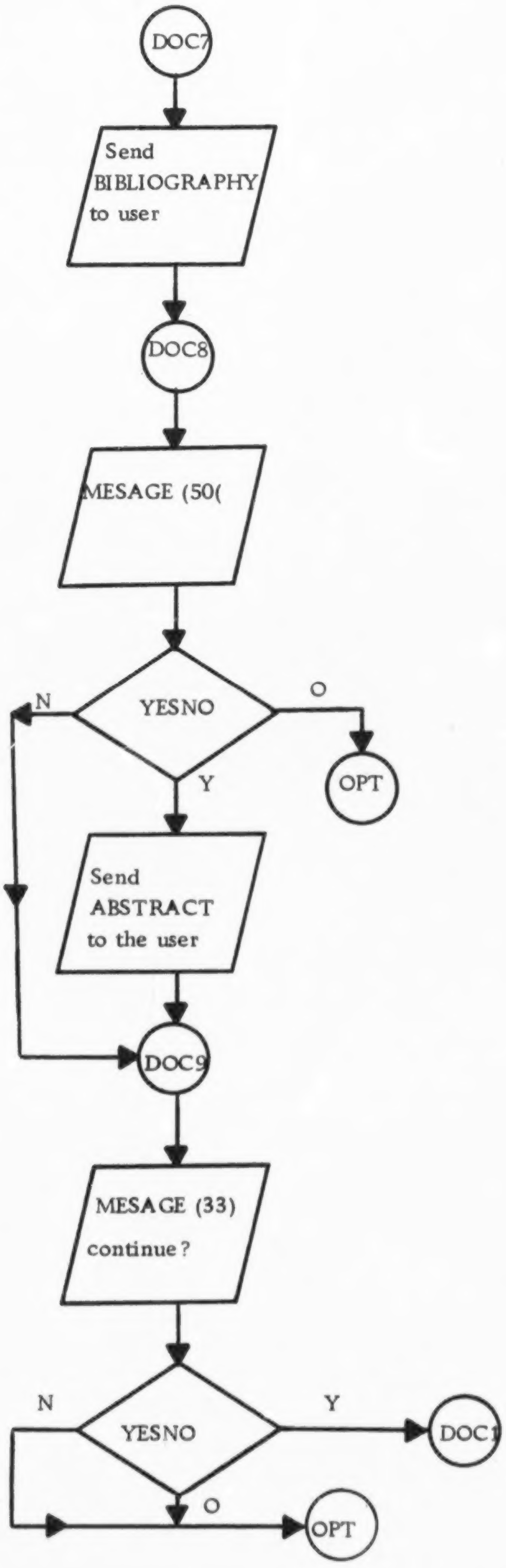

DOC7
Send BIBLIOGRAPHY to user
DOC8
MESAGE (50(
YESNO
N
O
Y
OPT
Send ABSTRACT to the user
DOC9
MESAGE (33) continue?
YESNO
N
Y
DOC1
O
OPT

Figure VI. 8 (Concluded)

In the normal course of events, the case of the "tyro" user or the experienced researcher without any particular need to manipulate a retrieval in detail, a retrieval of documents is made immediately after the query concept vector is built. However, in the case of considerable query vector manipulation, the user may wish to perform a new retrieval at some other point. The ability to perform a retrieval at will is a quite powerful research tool, permitting the user to immediately test out the results of a particular modification to the query vector. Figure VI.9 presents the logic of RET.

WRD (6410): See and/or modify the words forming the present query.

As in the RET option (above), WRD is used in the detailed manipulation of the query. Words forming the query may be dealt with in several ways:

a) words may be added to the query
b) words may be deleted from the query

All words to be added are put through the same stemming routines (morphological analysis) as the initial query words, with the result that no distinction is made between the original words and those added as far a effect on the query goes. In addition, when words are deleted from the query, their slots in the tables are not removed until an actual query concept vector is built just prior to a retrieval. Unless the deleted words are blanked and not completely removed, the user may forget which words he removed. Leaving blanks in place of the deleted word (with a character-count of zero), prevents any erroneous query information from being generated. Figure VI.10 illustrates the logic of WRD.

CLR (6540): Set all of the query concept vector to zero, the default empty condition. This option is used simply to clear out the query concept vector. Figure VI.11 illustrates the logic of this simple routine.

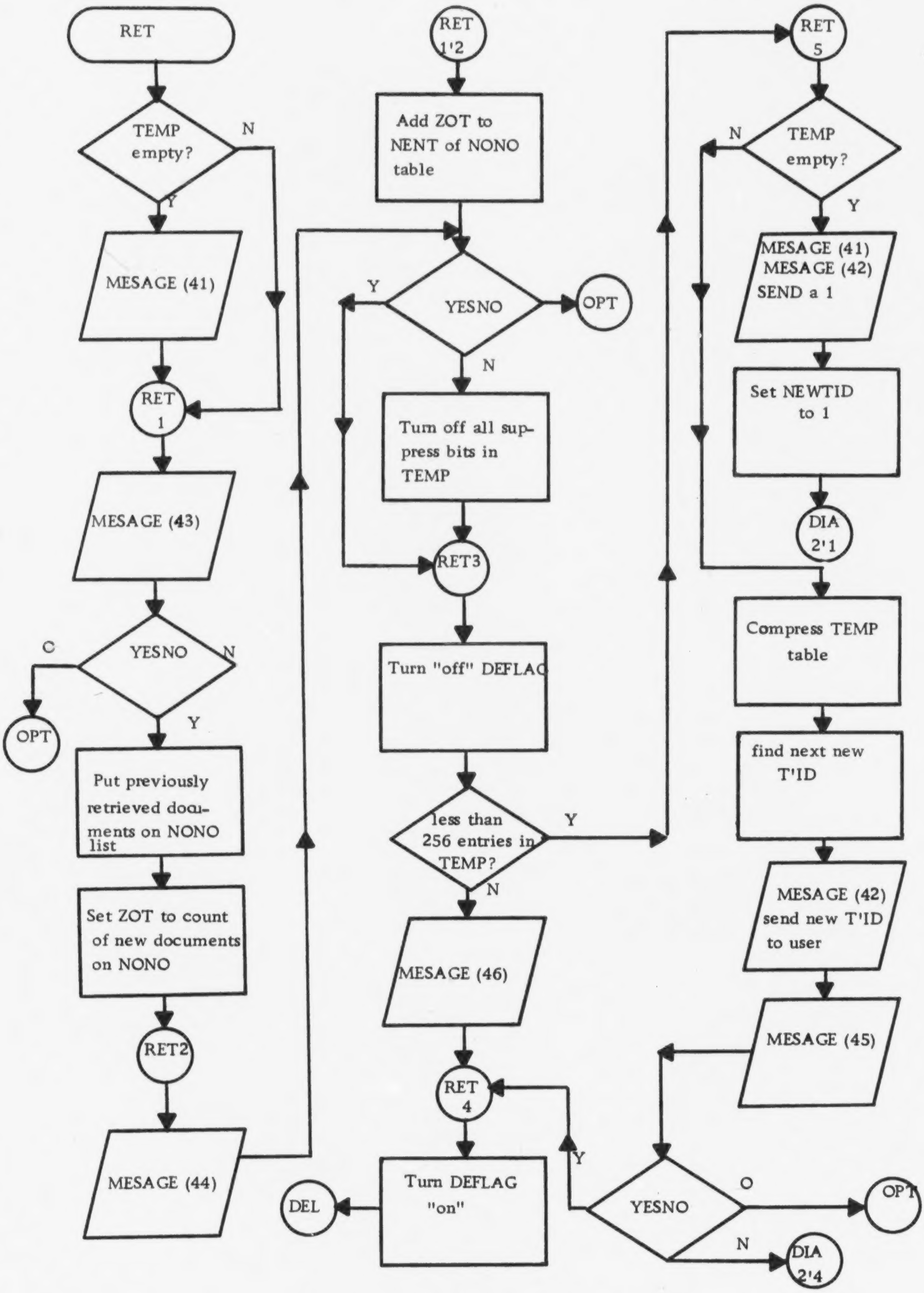

RET
TEMP empty?
N
Y
MESAGE (41)
RET 1
MESAGE (43)
YESNO
C
N
Y
OPT
Put previously retrieved documents on NONO list
Set ZOT to count of new documents on NONO
RET2
MESAGE (44)
RET 1'2
Add ZOT to NENT of NONO table
YESNO
Y
OPT
N
Turn off all suppress bits in TEMP
RET3
Turn "off" DEFLAG
less than 256 entries in TEMP?
Y
N
MESAGE (46)
RET 4
Turn DEFLAG "on"
DEL
RET 5
TEMP empty?
N
Y
MESAGE (41)
MESAGE (42)
SEND a 1
Set NEWTID to 1
DIA 2'1
Compress TEMP table
find next new T'ID
MESAGE (42) send new T'ID to user
MESAGE (45)
YESNO
Y
O
N
OPT
DIA 2'4

Figure VI. 9. RET Flowchart

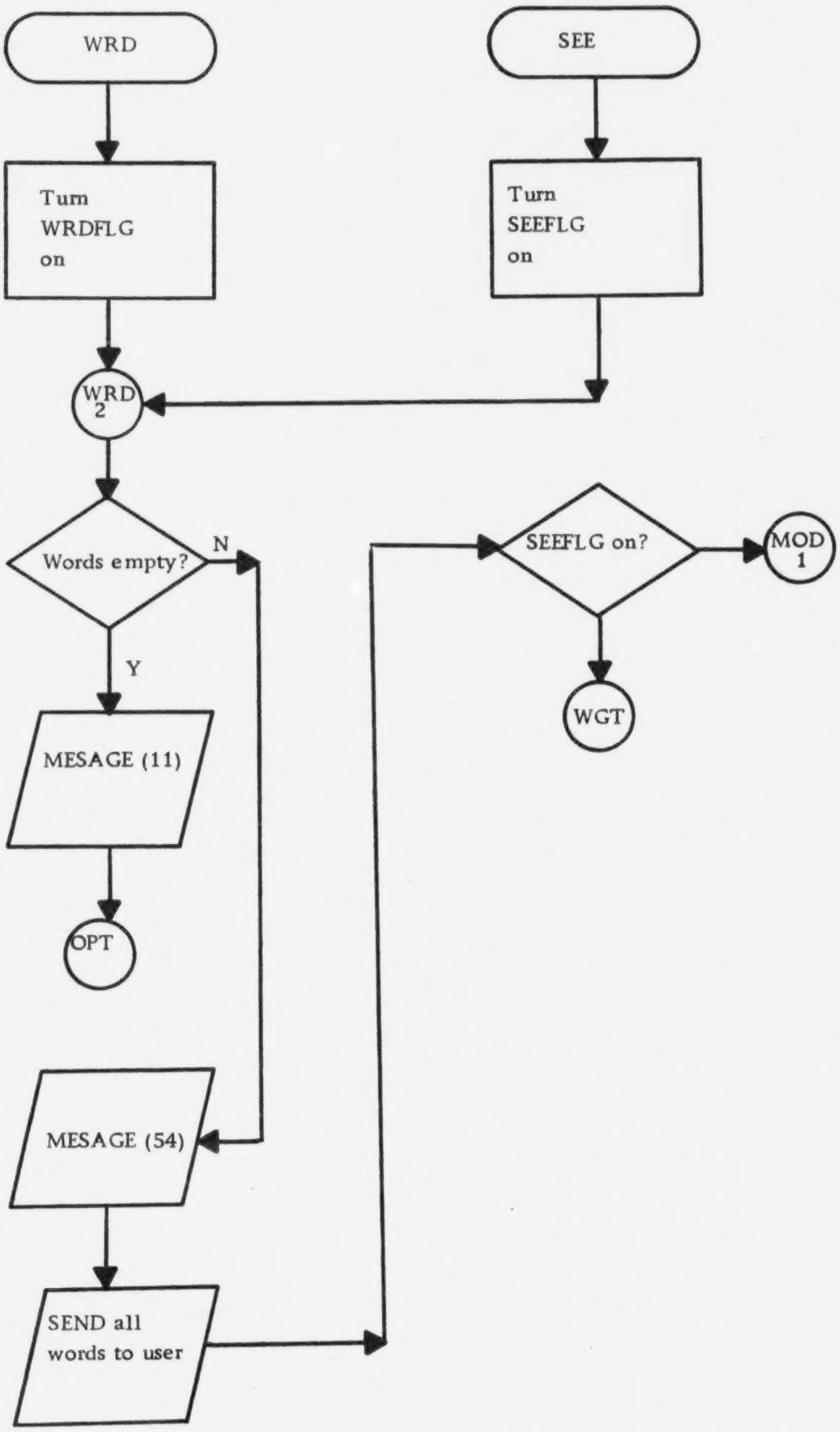

WRD
SEE
Turn WRDFLG on
Turn SEEFLG on
WRD 2
Words empty?
N
Y
SEEFLG on?
MOD 1
WGT
MESAGE (11)
OPT
MESAGE (54)
SEND all words to user

Figure VI.10. WRD & SEE Flowchart

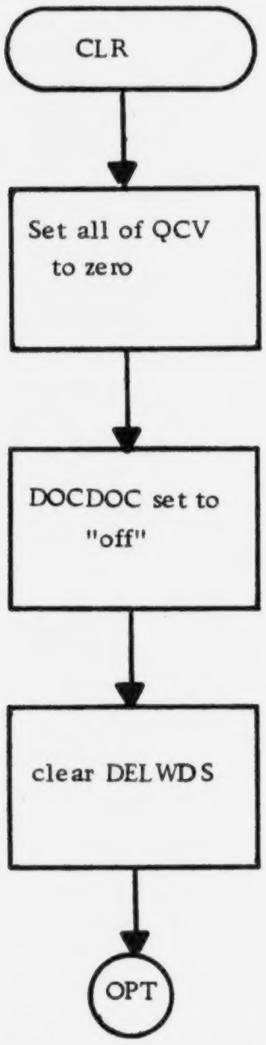

CLR
Set all of QCV to zero
DOCDOC set to "off"
clear DELWDS
OPT

Figure VI. 11. CLR Flowchart

DEL (6670): Delete documents from the temporary file.

As a result of a query, a temporary "file" of documents is built. However, design limitations prevent there from being more than 256 places in this table. User function DEL is used to remove specified documents from the TEMP table to make space for further retrievals. Figure VI.12 references function DEL.

MOD (7050): Modify the query function.

The MOD function is used to drive all the other query modifying options in the system. MOD is ordinarily called automatically by the system as the user interacts with the program to obtain information about documents retrieved. At his option, the user may elect to call the modification routines himself, assuming direct control over their operation. Routines called directly or indirectly by MOD are the following:

a) WGT
b) WRD
c) DDC

Figure VI.13 is the flowchart for MOD.

SEE (6440): The SEE function is an entry point in WRD.

CON (8320): Examine the document concept vectors for selected documents. The serious information retrieval researcher may be interested in seeing the actual document concept vectors of documents just retrieved, in order to determine which concepts and weights each document contributed to the formation of document-document correlation. Figure VI.14 is the flowchart for CON.

ENDF (8830): End-query function.

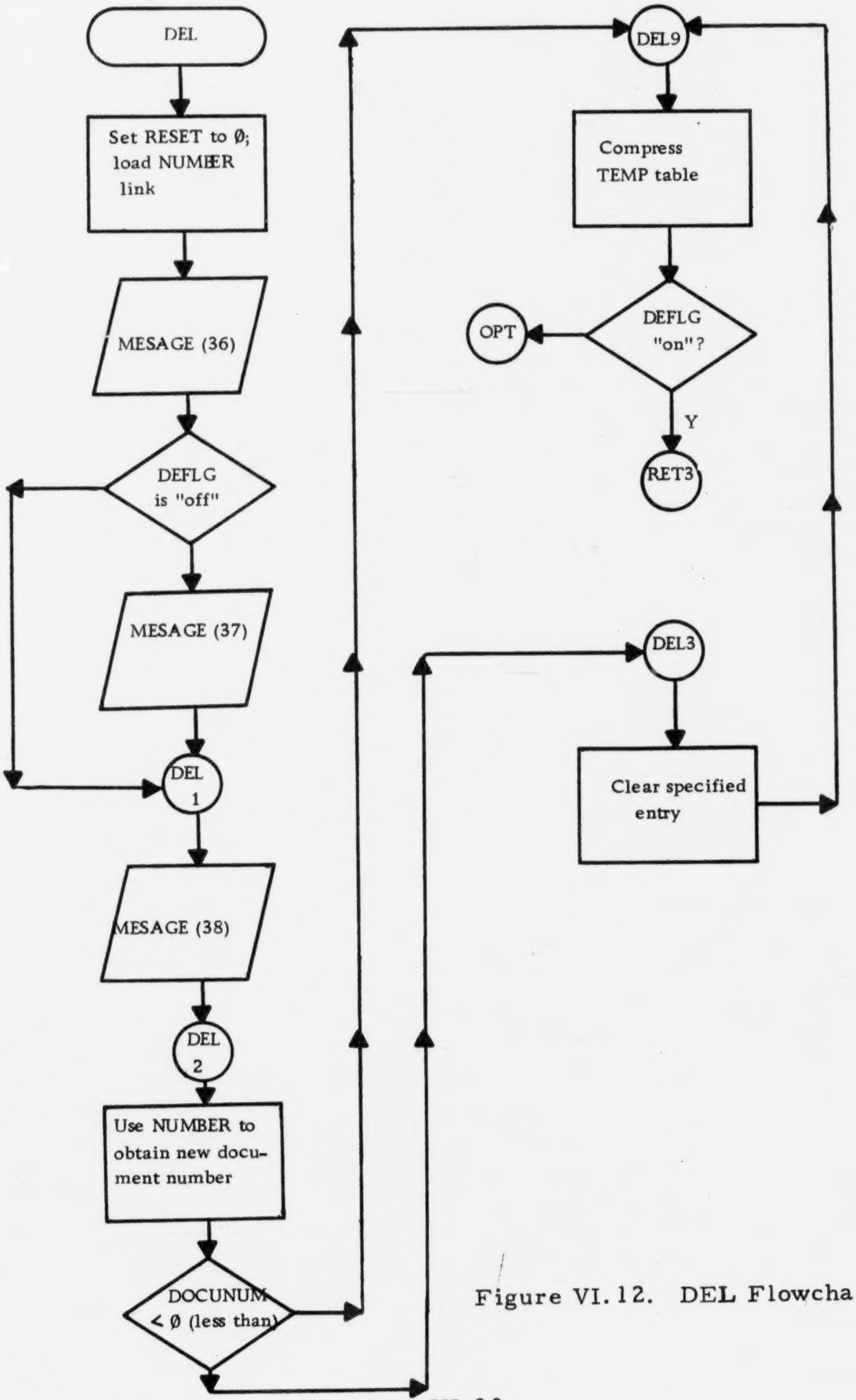

DEL
Set RESET to 0; load NUMBER link
MESAGE (36)
DEFLG is "off"
MESAGE (37)
DEL 1
MESAGE (38)
DEL 2
Use NUMBER to obtain new document number
DOCUNUM < 0 (less than)
DEL9
Compress TEMP table
DEFLG "on"?
OPT
Y
RET3
DEL3
Clear specified entry

Figure VI. 12. DEL Flowchart

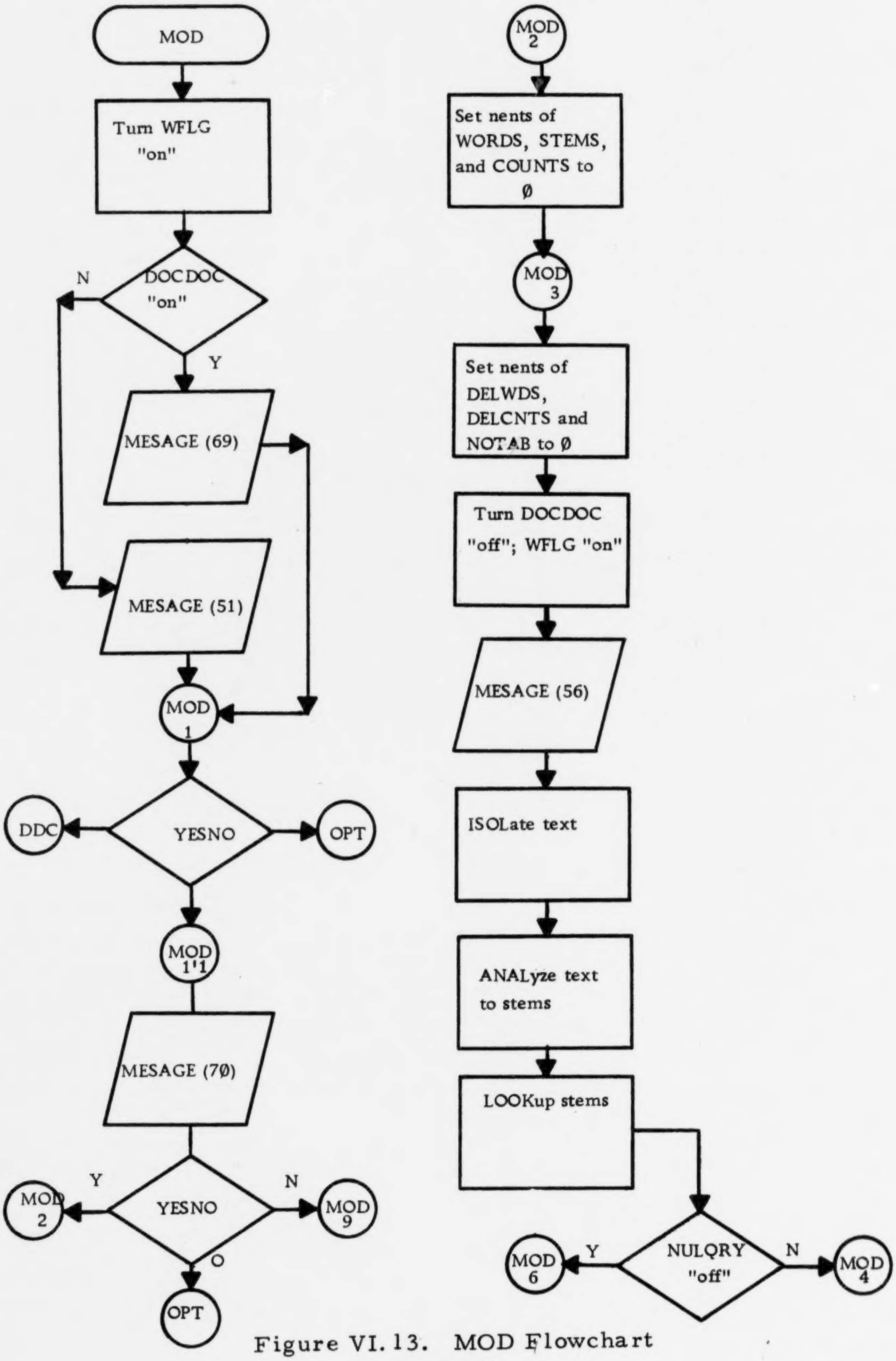

MOD
Turn WFLG "on"
DOCDOC "on"
N
Y
MESAGE (69)
MESAGE (51)
MOD 1
DDC
YESNO
OPT
MOD 1'1
MESAGE (70)
Y
N
YESNO
O
MOD 2
MOD 9
OPT
MOD 2
Set nents of WORDS, STEMS, and COUNTS to 0
MOD 3
Set nents of DELWDS, DELCNTS and NOTAB to 0
Turn DOCDOC "off"; WFLG "on"
MESAGE (56)
ISOLate text
ANALyze text to stems
LOOKup stems
NULQRY "off"
Y
N
MOD 6
MOD 4

Figure VI.13. MOD Flowchart

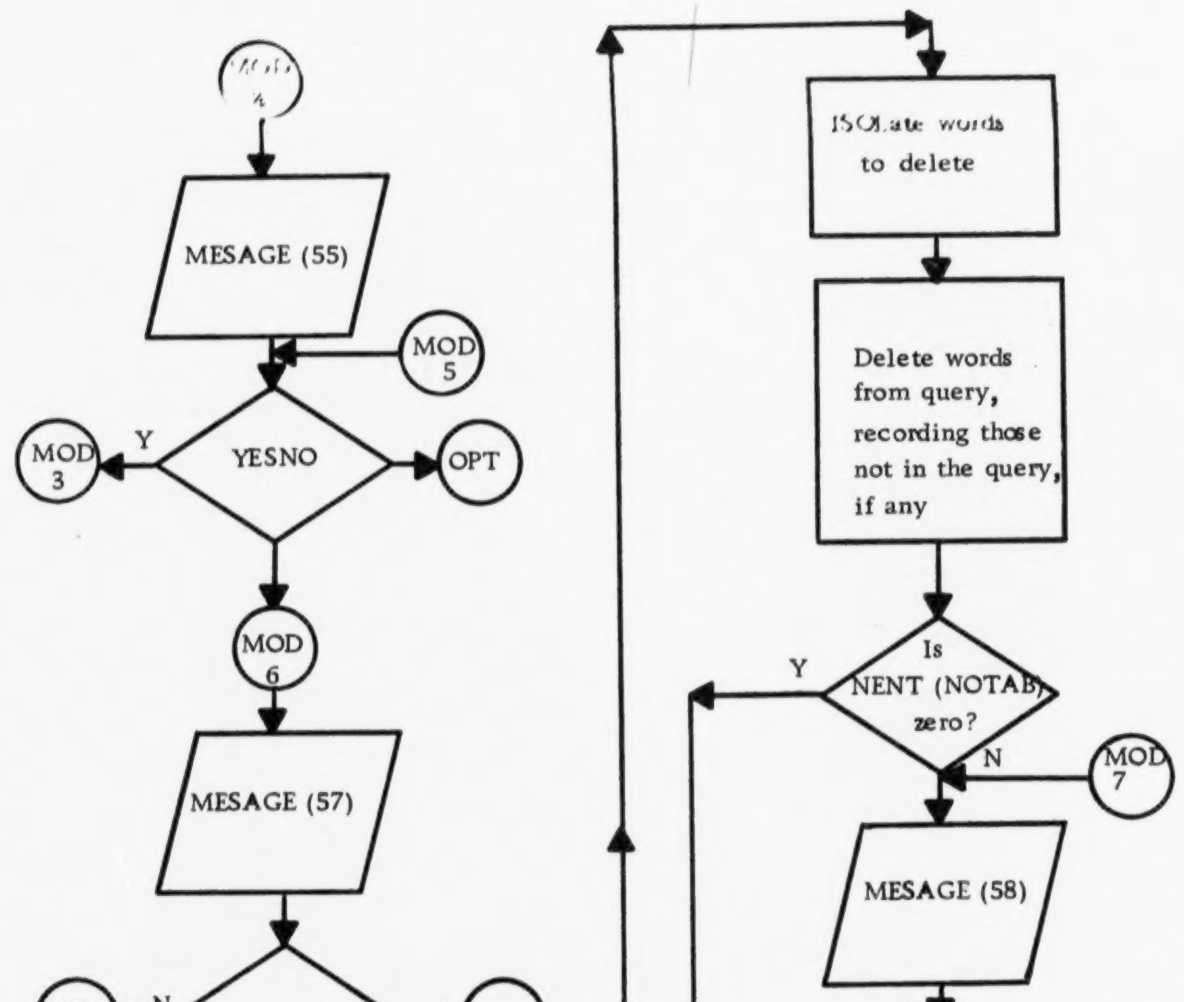

MOD 4
MESAGE (55)
MOD 5
MOD 3
Y
YESNO
OPT
MOD 6
MESAGE (57)
MOD 1'1
N
YESNO
OPT
Y
Turn DOCDOC "off"; WFLG "on"
MESAGE (56)
Set NENT (NOTAB) to Ø
ISOLate words to delete
Delete words from query, recording those not in the query, if any
Y
Is NENT (NOTAB) zero?
N
MOD 7
MESAGE (58)
SEND "bad" words to user
Reset NENT of NOTAB to zero
MOD 8

Figure VI. 13. (Continued)

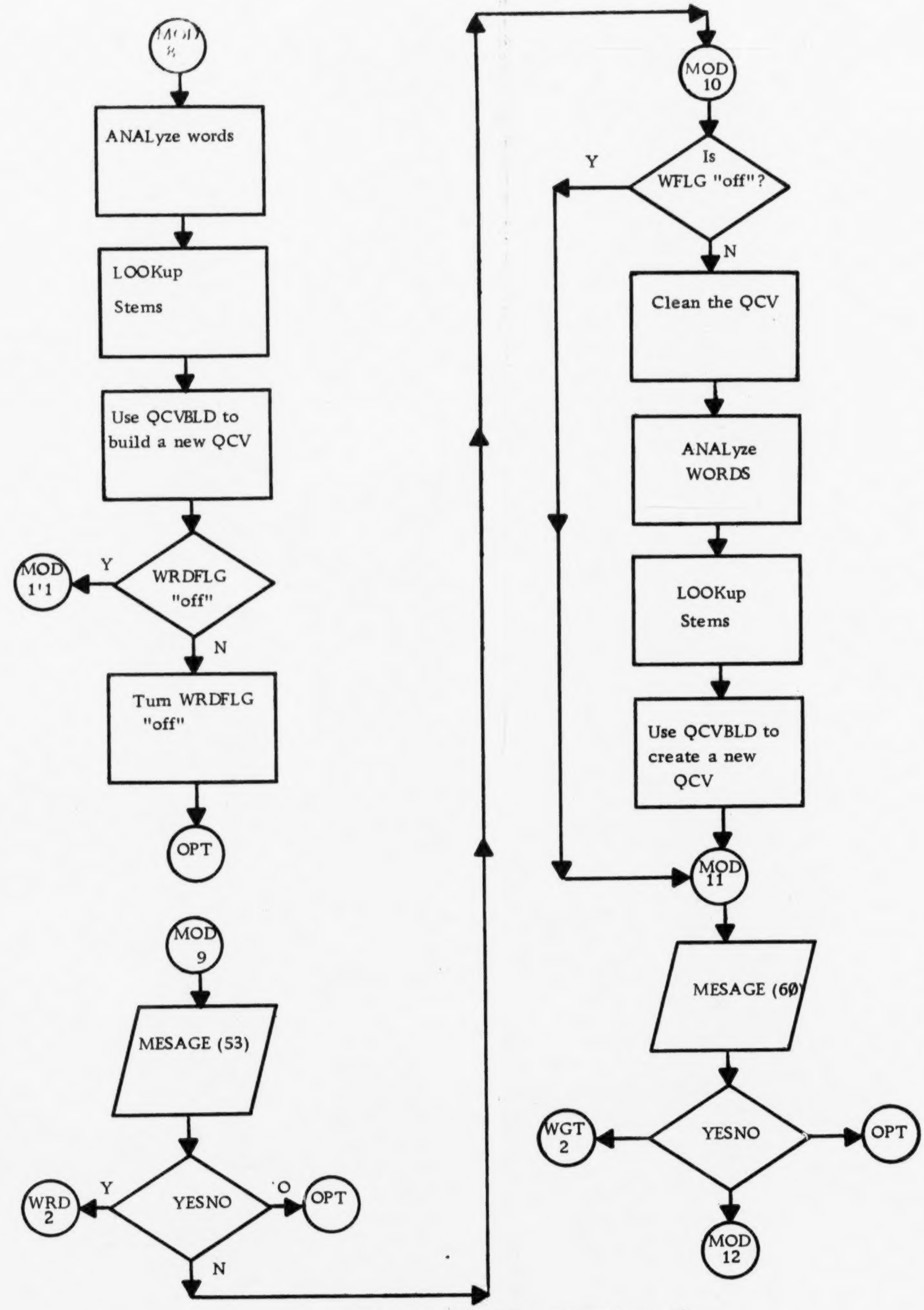

MOD 8
ANALyze words
LOOKup Stems
Use QCVBLD to build a new QCV
WRDFLG "off"
Y
MOD 1'1
N
Turn WRDFLG "off"
OPT
MOD 9
MESAGE (53)
YESNO
Y
WRD 2
O
OPT
N
MOD 10
Is WFLG "off"?
Y
N
Clean the QCV
ANALyze WORDS
LOOKup Stems
Use QCVBLD to create a new QCV
MOD 11
MESAGE (60)
YESNO
WGT 2
OPT
MOD 12

Figure VI.13. (Continued)

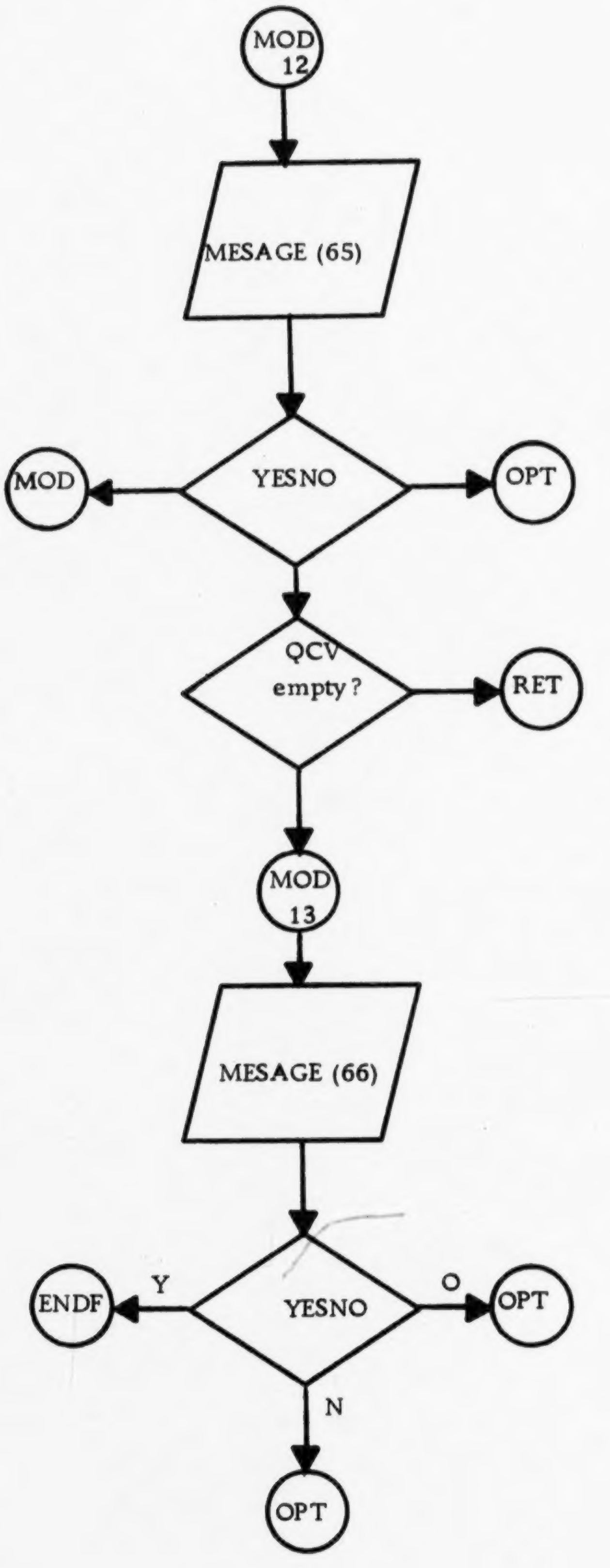

MOD 12
MESAGE (65)
MOD
YESNO
OPT
QCV empty?
RET
MOD 13
MESAGE (66)
ENDF
Y
YESNO
O
OPT
N
OPT

Figure VI. 13. (Concluded)

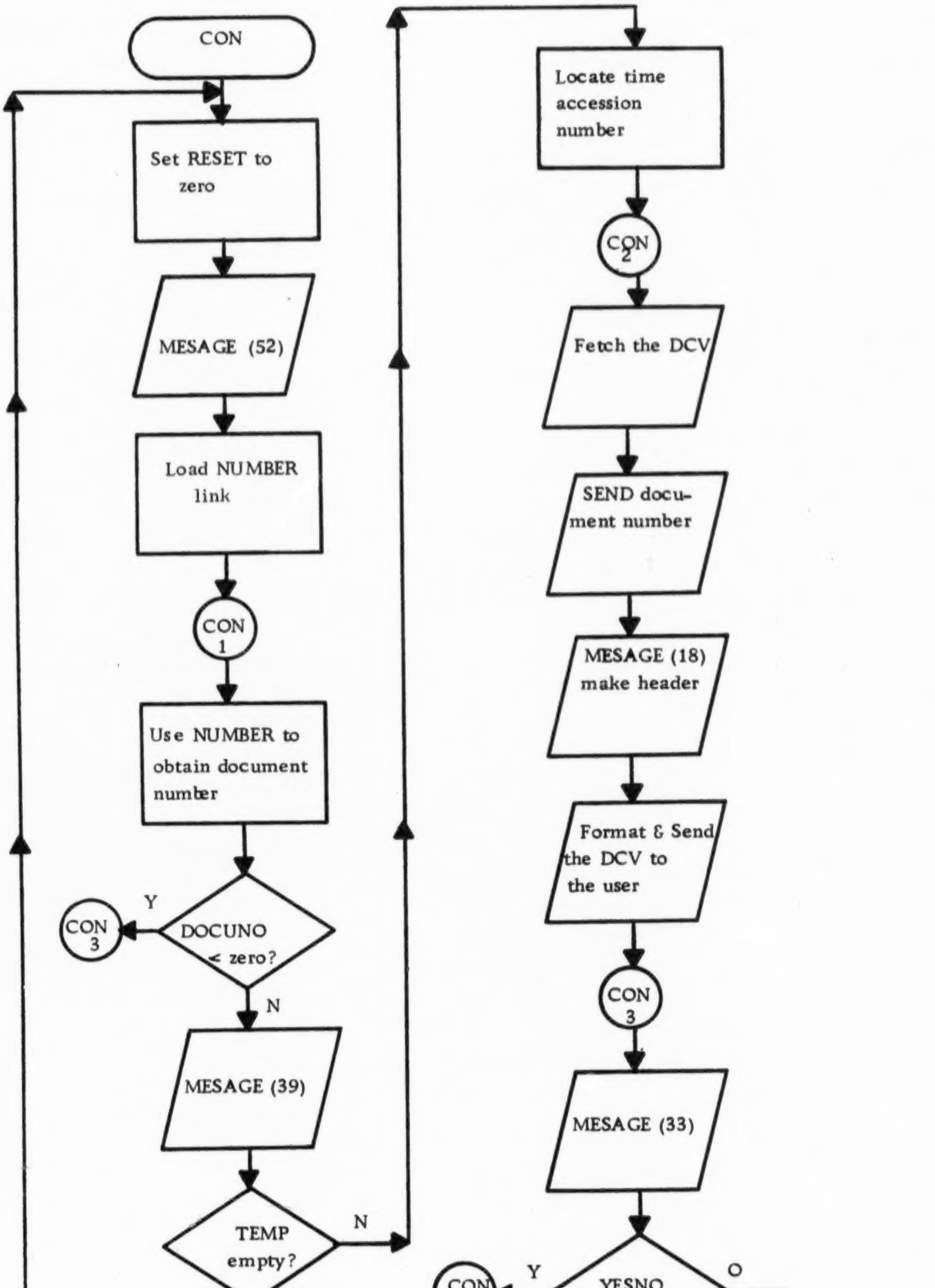

CON
Set RESET to zero
MESAGE (52)
Load NUMBER link
CON 1
Use NUMBER to obtain document number
Y
CON 3
DOCUNO < zero?
N
MESAGE (39)
TEMP empty?
N
Y
MESAGE (6)
Locate time accession number
CON 2
Fetch the DCV
SEND document number
MESAGE (18) make header
Format & Send the DCV to the user
CON 3
MESAGE (33)
Y
CON
YESNO
O
N
OPT

Figure VI. 14. CON Flowchart

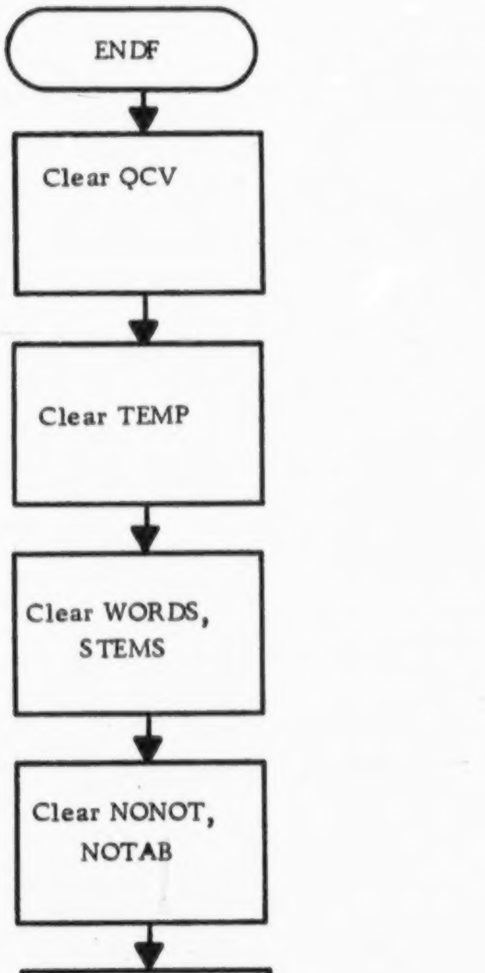

ENDF
Clear QCV
Clear TEMP
Clear WORDS, STEMS
Clear NONOT, NOTAB
Clear query-stem table, DELWRDS
Reset all modes, except TERSE
Reset all other flags
MESAGE (32)
DIA 5

Figure VI. 15. ENDF Flowchart

The use of the option ENDF results in the present query sequence being killed, all relevant tables cleared to their initial states, and the entire contents of the present query lost.

BLD (13000): build a query concept vector.
Flowchart is Figure VI. 16.

BLD causes an explicit call to QCVBLD, which in turn processes the present contents of the query-stem table QUERST and produces the sparse query concept vector from which a retrieval may be made. Ordinarily the user need not concern himself with this function, but during the course of constant query manipulation it may be desirable to cause an explicit formation of a new query concept vector through the use of this function.

WGT (9230): modify directly the query concept vector.
Flowchart is Figure VI. 17.

The WGT function allows the user to modify directly the query concept vector, using a specified syntax. In this way an existing query may be changed or an entirely new one built from "scratch". When this function is called, a print-out is given of the present query concept vector as it stands before modification. The condition of "null query concept vector" is also recognized.

SKIM (21000): skim off entries from TEMP.
Flowchart is Figure VI. 19.

The SKIM function is a powerful tool used to remove entries from the temporary retrieval table (TEMP) after a retrieval has been performed. The syntax of input to this function is the same as for the WGT function, with the except that the "concept" field is ignored. All entries less than the specified value are removed from TEMP, the TEMP table is compressed, and a message is sent indicating how many entries are left after the SKIMming operation.

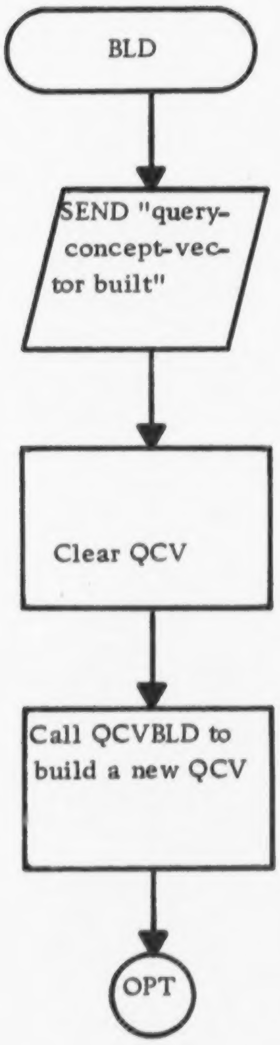

BLD
SEND "query-concept-vector built"
Clear QCV
Call QCVBLD to build a new QCV
OPT

Figure VI.16. BLD Flowchart

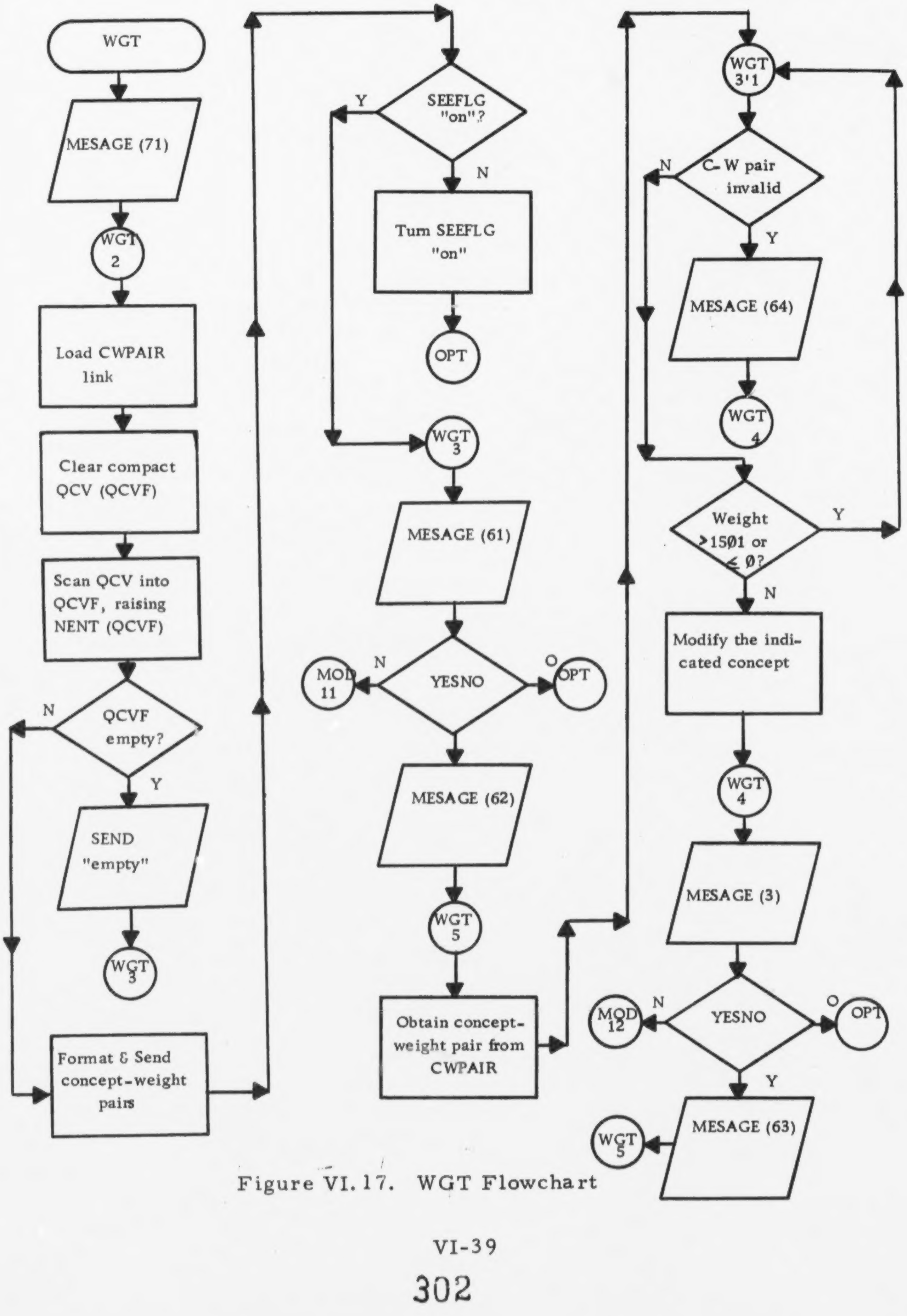

WGT
MESAGE (71)
WGT 2
Load CWPAIR link
Clear compact QCV (QCVF)
Scan QCV into QCVF, raising NENT (QCVF)
QCVF empty?
N
Y
SEND "empty"
WGT 3
Format & Send concept-weight pairs
SEEFLG "on"?
Y
N
Turn SEEFLG "on"
OPT
WGT 3
MESAGE (61)
YESNO
N
MOD 11
O
OPT
MESAGE (62)
WGT 5
Obtain concept-weight pair from CWPAIR
WGT 3'1
C-W pair invalid
N
Y
MESAGE (64)
WGT 4
Weight >1501 or ≤ 0?
Y
N
Modify the indicated concept
WGT 4
MESAGE (3)
YESNO
N
MOD 12
O
OPT
Y
MESAGE (63)
WGT 5

Figure VI.17. WGT Flowchart

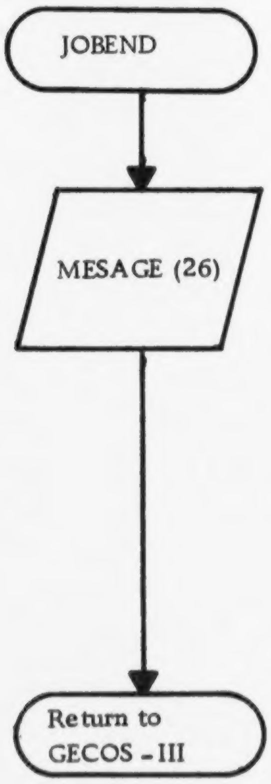

JOBEND
MESAGE (26)
Return to
GECOS - III

Figure VI. 18. Flowchart for JOBEND

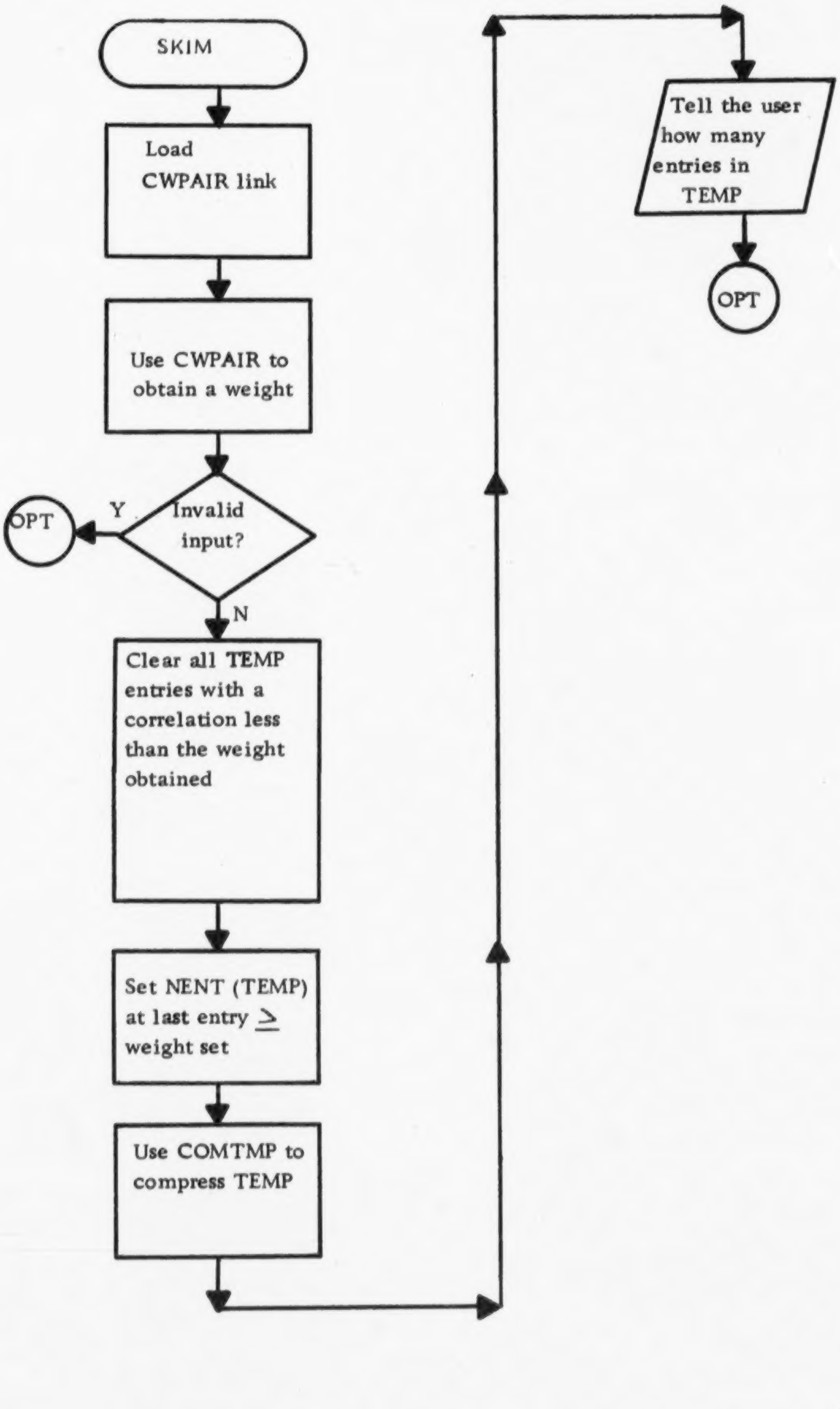

SKIM
Load CWPAIR link
Use CWPAIR to obtain a weight
Invalid input?
Y
OPT
N
Clear all TEMP entries with a correlation less than the weight obtained
Set NENT (TEMP) at last entry ≥ weight set
Use COMTMP to compress TEMP
Tell the user how many entries in TEMP
OPT

Figure VI. 19. SKIM Flowchart

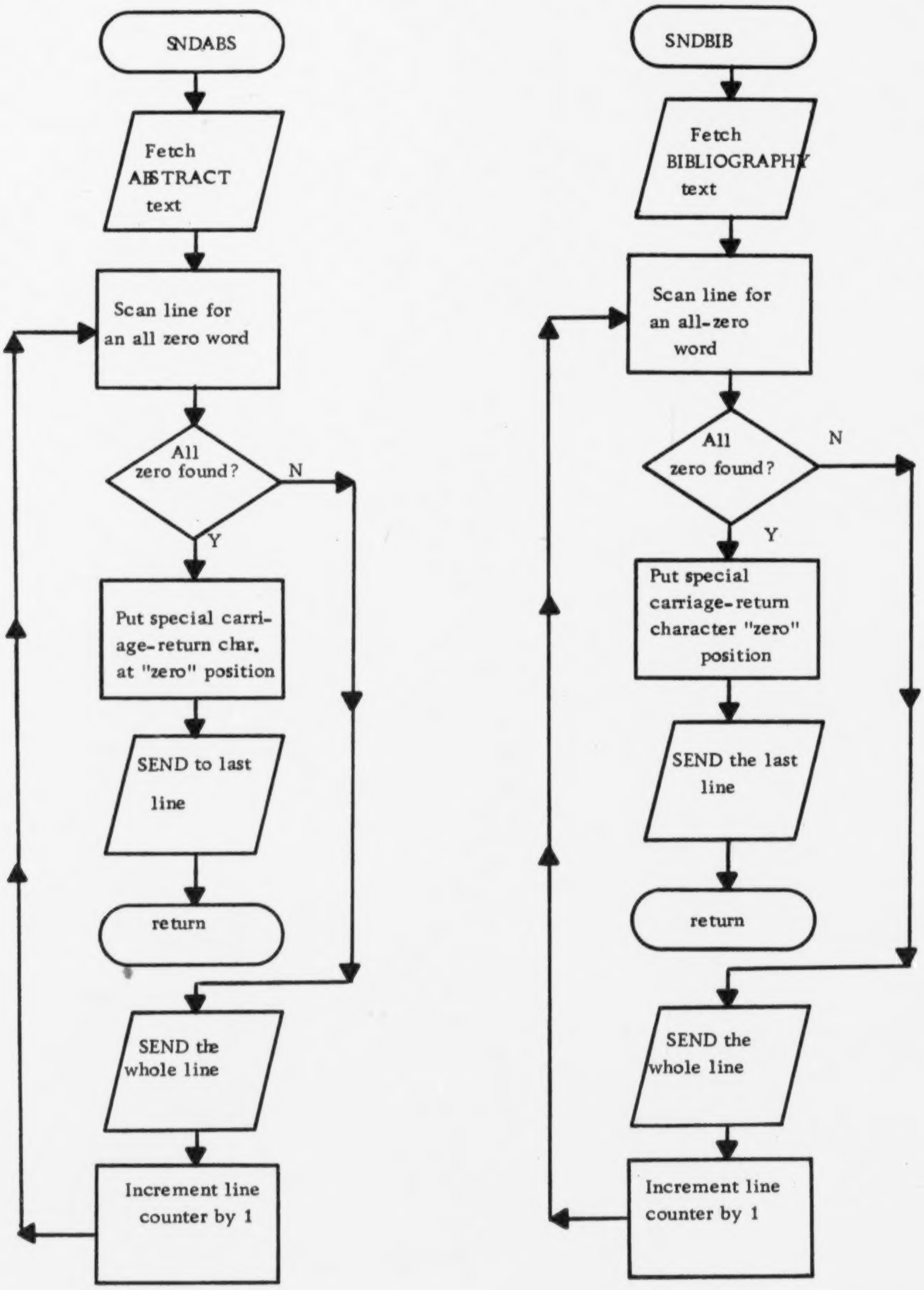

SNDABS
Fetch ABSTRACT text
Scan line for an all zero word
All zero found?
N
Y
Put special carriage-return char. at "zero" position
SEND to last line
return
SEND the whole line
Increment line counter by 1
SNDBIB
Fetch BIBLIOGRAPHY text
Scan line for an all-zero word
All zero found?
N
Y
Put special carriage-return character "zero" position
SEND the last line
return
SEND the whole line
Increment line counter by 1

Figure VI. 20. SNDABS; SNDBIB

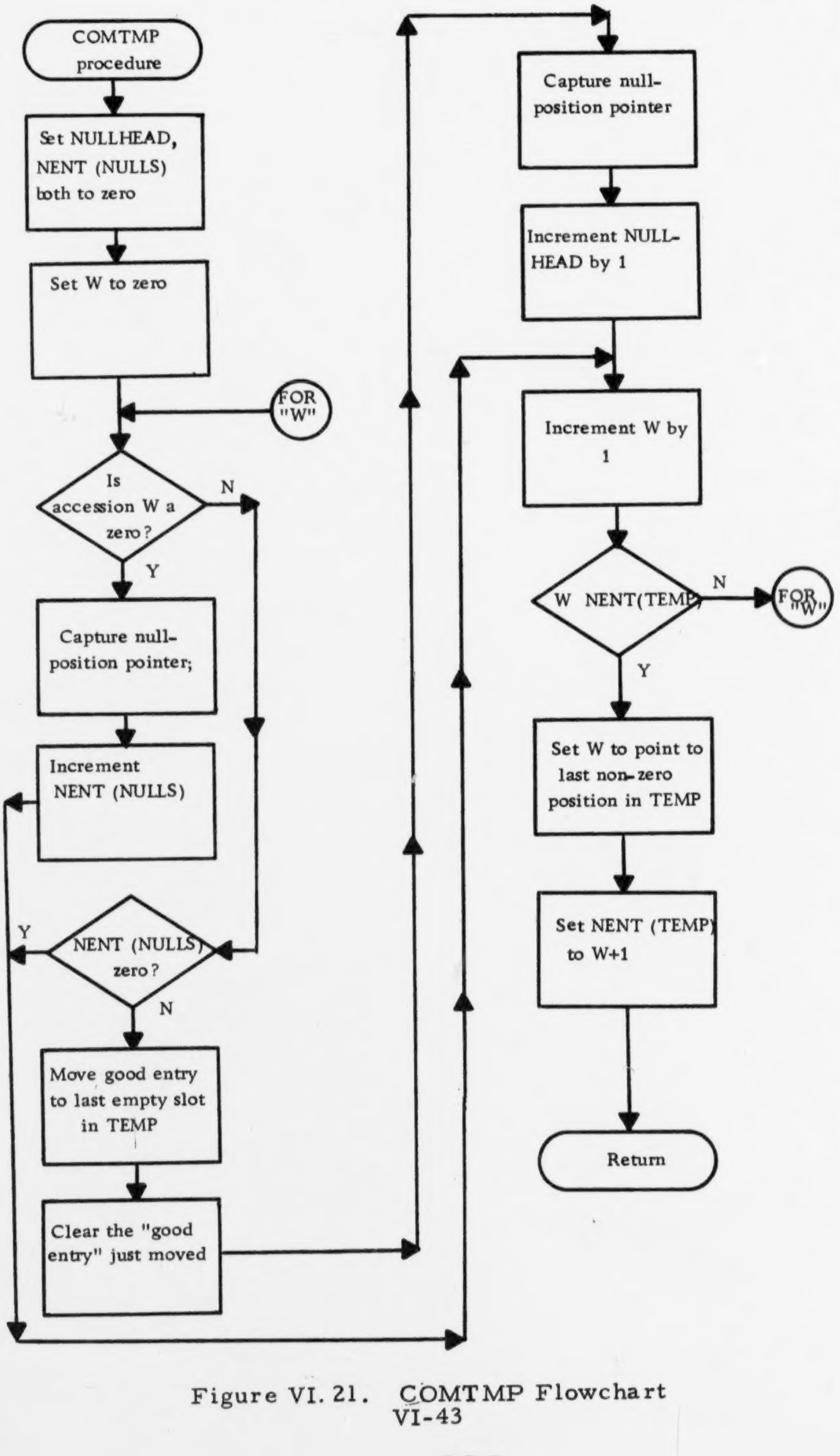

COMTMP procedure
Set NULLHEAD, NENT (NULLS) both to zero
Set W to zero
FOR "W"
Is accession W a zero?
N
Y
Capture null-position pointer;
Increment NENT (NULLS)
Y
NENT (NULLS) zero?
N
Move good entry to last empty slot in TEMP
Clear the "good entry" just moved
Capture null-position pointer
Increment NULL-HEAD by 1
Increment W by 1
W NENT(TEMP)
N
FOR "W"
Y
Set W to point to last non-zero position in TEMP
Set NENT (TEMP) to W+1
Return

Figure VI. 21. COMTMP Flowchart

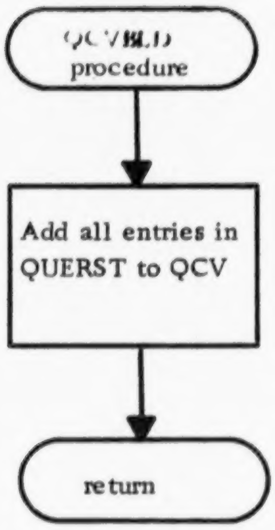

QCVBLD procedure
Add all entries in QUERST to QCV
return

Figure VI. 22. QCVBLD Flowchart

VI. 2. 3 Overlaid Portions of the On-Line System

VI. 2. 3. 1 Design Philosophy. In order to conserve valuable core space, the less frequently used portions of the on-line system have been segmented into overlays or LINKS in terms of GECOS-III. Each LINK is loaded from system file H* as needed, (determined by program logic) and executed. Requests are made through the GECOS-III library routine LLINK ("load-link"). No one of the program links is particularly large by itself, but the sum of all the links, if loaded all into core at the same time, would add somewhere in the vicinity of 10,000 extra words to the core requirements of the system at run time.

GECOS-III builds LINKS during the first part of the GELOAD activity if $ LINK control cards are present in the object stream. As new LINKS are encountered, those previously used areas of core which are to be overlaid are written out on file H* in system loadable format (essentially core-image form). After all LINKS are processed, the main segment and file-control-blocks, if any, are also written out. The result is that at run time (actual program execution time) file H* contains a complete copy of the core-image. If H* is assigned to tape disk pack or perm file, subsequent executions of the program can request a reload of the core-image, bypassing the entire first phase of the GELOAD activity. In terms of the user, utilizing H* as the loadable file for the program eliminates a needless delay of somewhat over a minute of elapsed time waiting at the terminal for the dialogue to start. Figure VI. 26 is an illustration of the deck set-up used when running the system from H*.

A call to LLINK does not by itself execute the chosen link. Instead, the program link is called into core and control is returned to the program which issued the LLINK call in the first place. During building of the links, overlays are treated as non-fatal-error undefined symbols until they are defined later on. Thus, calls to routines in links are exactly the same as calls to other procedures or functions. Caution must be exercised, however, to assure that the correct link is loaded at the time a procedure is called, because erroneous program calls could result. For this reason, the Dialogue Processor reloads each link as it is needed. Program links in this system are neither reentrant nor serially reusable, as they all modify their environment to some extent.

VI. 2. 3. 2 ISOLate. A small but important task which must be performed by the on-line system is the careful first-phase processing of raw textual material typed in by the terminal user. Procedure ISOLate handles this job.

Input text is processed character-by-character in accordance with a prestored character state-table. The entire character set available for the machine is broken up into four classes.

1) legal characters, i. e., those which may be used to form a text word
2) break characters, i. e., those which are word boundaries
3) garbage characters, i. e., those which are entirely ignored by ISOLate
4) end-of-text, which terminates further use of ISOLate unless an end-of-text character is encountered in the input stream, ISOLate continues to process and request new input text until the maximum number of text words is reached (set by a "define"). The current implementation allows a maximum of 100 (one hundred) sequential text words. Error recovery from exceeding this maximum is automatic; the terminal user is informed of the error condition, and the last word processed is sent to the terminal.

Calling parameters for ISOLate are the following:

1) input: none
2) output
   a) WORDS: table to receive the data from the user
   b) COUNTS: table, addressed in parallel with WORDS, to receive the count of legal characters in the corresponding entry in WORDS

In ISOLate data are always added to WORDS from a base address of NENT(WORDS) up to the maximum limit. Therefore, it is the responsibility of the calling routine to manipulate NENT(WORDS) as desired. Both

WORDS and COUNTS are formal tables in ISOLate and must have actual storage assigned in the calling program. The maximum legal word length (text words) for this version of the implementation is 18 (eighteen) characters, because of space limitations in the main segment of the processor. This limit could easily be raisde. ISOLate is taken from the micro-concordance generator.

VI.2.3.3 ANALyze. One of the key concepts underlying the whole retrieval system, both batch and on-line, is the idea of language normalization, i.e., the reduction of text words to common stems by eliminating suffixes. The algorithm for ANALyze is taken with as little modification as possible, considering that it must be used in an on-line environment, from the concordance generator (c.f. Section V.4). The chief difference between the on-line and batch versions of the program is that the on-line version processes the WORDS and COUNTS tables generated by ISOLate, rather than processing each text word as it is isolated from input, as in the micro-concordance program. This division of labor allows ANALyze to be quite independent of ISOLate, a distinct advantage if another language besides English is to be used in the system at some future date. Also, different methods of language normalization could be tried in an experimental environment, without the necessity of reprogramming the entire on-line system. Specifically, syntax analysis might be attempted, as well as other stemming algorithms.

In ANALyze the input text present in the WORDS table is subject to two different processes:

1) common-word elimination
2) stemming

These processes are described in detail below.

Common-word elimination is needed in the language normalization process to prevent needless analysis of noncontent text matter. Two tables of words are used:

1) common-words with six or fewer characters

2) common-words with seven or more characters (upper bound in twelve characters)

The tables are divided in the manner described to speed up the lookup process. Procedure HUNT1 is used to binary-search the common-word table of one machine-word in width, while HUNT2 is used on the two-word-wide table.

Stemming is carried out in a right-to-left fashion, removing the longest suffixes first. At no time is the stem of a word reduced to fewer than four characters. In fact a candidate-stem of four or fewer characters which is not a common word is automatically retained as a legel stem. Singleton characters are considered "garbage," and are also eliminated.

At all time the WORDS and COUNTS tables are kept in parallel by continual compression of rejected words; the STEMS table the output area of ANALyze, is also kept in parallel with WORDS and COUNTS. Calling parameters for ANAL are the following:

1) input
    a) WORDS
    b) COUNTS
2) output
    a) WORDS
    b) COUNTS
    c) STEMS

Tables WORDS and COUNTS are both input and output parameters so that the JOVIAL compiler generated code will allow the necessary updating of these areas without objecting to the syntactic construction. Besides, those tables are indeed both input and output area for the routine. ANALyze is taken directly from the micro-concordance generator.

VI. 2. 3. 4 LOOK (Dictionary Lookup in the Stem-Dictionary)

The last phase of the analysis portion of the Dialogue Processor is the LOOK link. Included in this overlay are four routines:

1. LOOK: the driver program
2. RDBUCK: the random-access disc reader
3. HUNT2A: a specialized version of HUNT2; returns the search pointer to the last table area searched; used to find stems in the table of contents, as described below.
4. HUNT2B: another specialized version of HUNT2; differs only in that it uses the first 12 bytes of a 36-byte-wide table as the search key.

The discussion below will bear most heavily on LOOK and RDBUCK, with some brief mention of the differences between HUNT2A/B and HUNT2 from which they are derived.

The purpose of LOOK is to determine if a stem, as returned by ANALyze is indeed a "content-stem". "Content stems" are defined as having the highest V_c value (c.f. V.5 for definition). As described in Section V.6, and especially in more detail in Section V.6.2.3, each content stem has associated with it a set of four concept-weight pairs. Since these pairs are needed to perform the retrieval, incoming stems not only must be looked up in the dictionary, but the pairs associated with these stems must be copied from the dictionary into the query stem preparation area. LOOK performs all the above functions.

It was abundantly clear that core limitations would prevent the inclusion of the entire stem dictionary in LOOK itself, so a different scheme had to be worked out. A two-level search was chosen as the fastest and most economical solution. An incoming stem is first looked up against a list of 500 stems obtained from the stem-dictionary loader DILOAD. Each of these entries represents the last stem in a disc bucket. There are 500 disc buckets of 10 entries each. HUNT2A is used to determine if the candidate stem falls in the range of the dictionary or not. It is, of course, possible to have a stem which is further down the alphabet than the last end-of-bucket indicator, in which case it obviously cannot be found in the dictionary. Such stems are immediately rejected. Two other possible conditions exist.

First, if a stem exactly matches any end-of-bucket entry, searching of the indicated bucket is irrelevant. The second matter concerns the desire not to reload the same disc bucket into core. Such a situation can and will occur when several stems in a row are close in the alphabet. Again, the algorithm takes this characteristic of alphabetized data into account: the last bucket read into core is kept track of, to avoid rereading the same data. In this way, during any one search cycle each bucket of disc is read one time only.

LOOK processes the STEMS table produced by the previous phases of analysis, keeping track of those words, the stems of which cannot be found in the stem dictionary. The system user is notified of these nondictionary words present in his query input, and, through MOD or WRD, has the opportunity to add or delete words from his query. Nondictionary entries are removed from the WORDS table, as well as from the STEMS table, so that valuable table space is not wasted. In the case that none of the user's query words were valid retrieval keys from the document collection, he is so informed and is encouraged to start another query. Figure VI.23 is the flowchart for the LOOK procedure.

RDBUCK is the random-access disc-file driver for the stem-dictionary. Use of this routine is simple, although several unforseen problems cropped up during the testing phases. Entry into RDBUCK is through a normal JOVIAL procedure call with one input parameter, the block number, and one output parameter, the bucket of content stems.

As originally conceived, RDBUCK read a disc bucket of 240 machine-words or 40 entries of 6 machine-words each. However, GEFRC has steadfastly refused to allow more than 64 machine-words to be read from disc in any one block, forcing a change to 60-word blocks of 10 entries each. In this case the table of contents in LOOK

1) legal characters, i.e., those which may be used to form a text word
2) break characters, i.e., those which are word boundaries
3) garbage characters, i.e., those which are entirely ignored by ISOLate.
4) end-of-text, which terminates further use of ISOLate

unless an end-of-text character is encountered in the input stream, ISOLate continues to process and request new input text until the maximum number of text words is reached (set by a "define"). The current implementation allows a maximum of 100 (one hundred) sequential text words. Error recovery from exceeding this maximum is automatic; the terminal user is informed of the error condition, and the last word processed is sent to the terminal.

Calling parameters for ISOLate are the following:

1) input: none
2) output
   a) WORDS: table to receive the data from the user
   b) COUNTS: table, addressed in parallel with WORDS, to receive the count of legal characters in the corresponding entry in WORDS

In ISOLate data are always added to WORDS from a base address of NENT(WORDS) up to the maximum limit. Therefore, it is the responsibility of the calling routine to manipulate NENT(WORDS) as desired. Both WORDS and COUNTS are formal tables in ISOLate and must have actual storage assigned in the calling program. The maximum legal word length (text words) for this version of the implementation is 18 (eighteen) characters, because of space limitations in the main segment of the processor. This limit could easily be raised. ISOLate is taken from the micro-concordance generator.

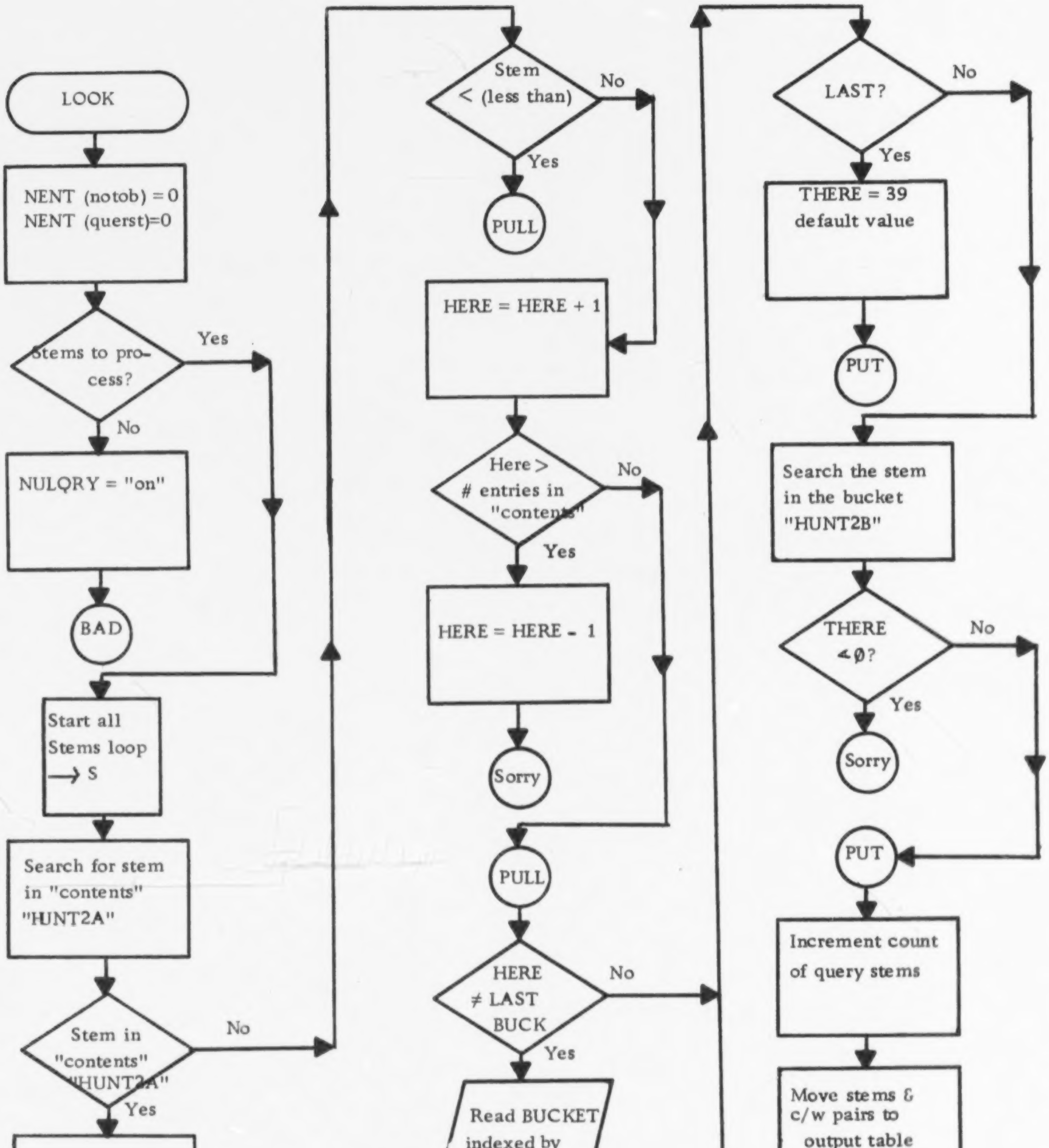

LOOK
NENT (notob) = 0
NENT (querst)=0
Stems to process?
Yes
No
NULQRY = "on"
BAD
Start all Stems loop → S
Search for stem in "contents" "HUNT2A"
Stem in "contents" "HUNT2A"
No
Yes
Last ← true
PULL
Stem < (less than)
No
Yes
PULL
HERE = HERE + 1
Here > # entries in "contents"
No
Yes
HERE = HERE - 1
Sorry
PULL
HERE ≠ LAST BUCK
No
Yes
Read BUCKET indexed by HERE
LAST BUCK = HERE
LAST?
No
Yes
THERE = 39 default value
PUT
Search the stem in the bucket "HUNT2B"
THERE < 0?
No
Yes
Sorry
PUT
Increment count of query stems
Move stems & c/w pairs to output table
"TEST S" check loop

Figure VI. 23. LOOK Flowchart

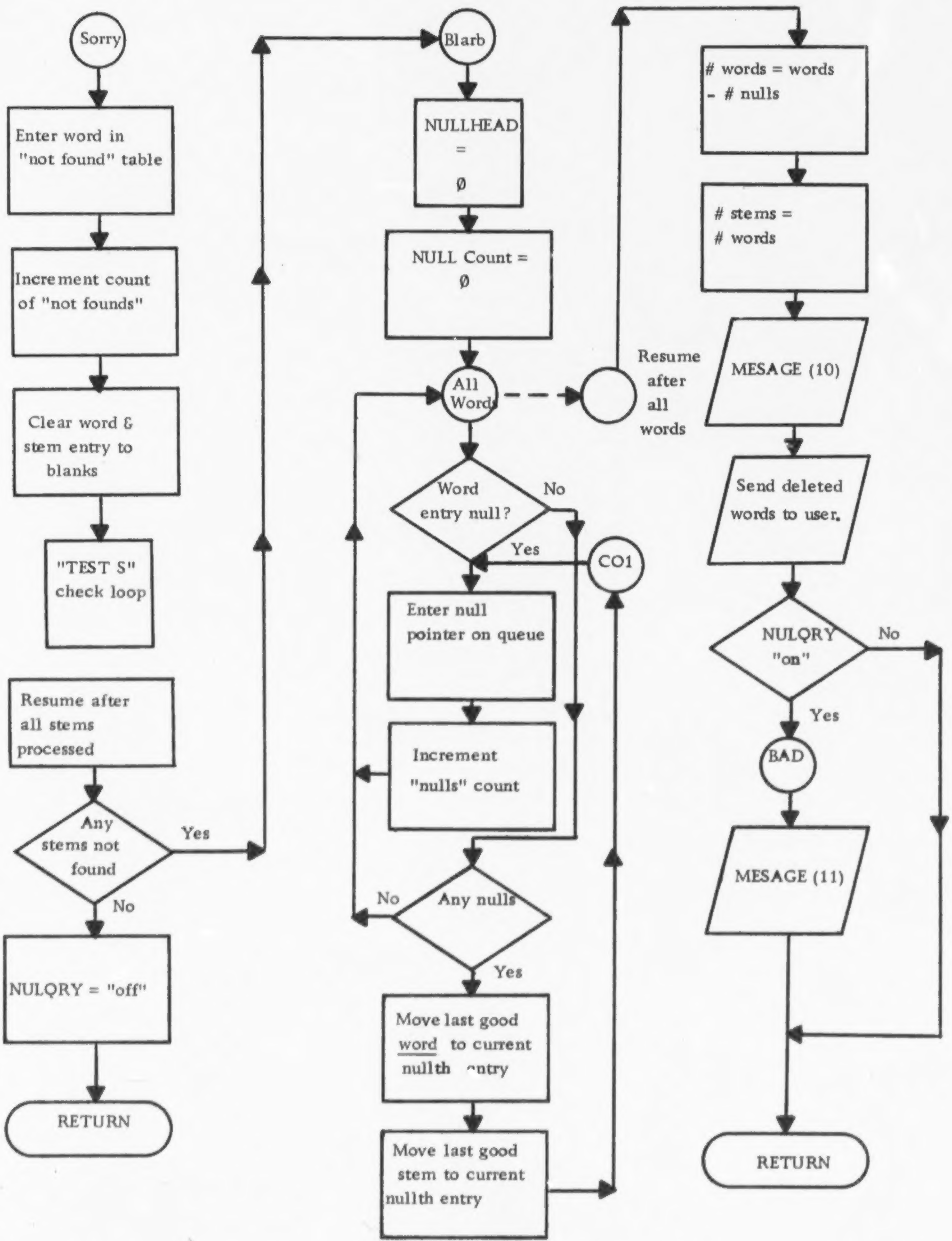

Sorry
Enter word in "not found" table
Increment count of "not founds"
Clear word & stem entry to blanks
"TEST S" check loop
Resume after all stems processed
Any stems not found
Yes
No
NULQRY = "off"
RETURN
Blarb
NULLHEAD = Ø
NULL Count = Ø
All Words
Resume after all words
Word entry null?
No
Yes
CO1
Enter null pointer on queue
Increment "nulls" count
Any nulls
No
Yes
Move last good word to current nullth entry
Move last good stem to current nullth entry
# words = words - # nulls
# stems = # words
MESAGE (10)
Send deleted words to user.
NULQRY "on"
No
Yes
BAD
MESAGE (11)
RETURN

Figure VI. 23. (Concluded)

```
50''********** DICTIONARY LOOKUP .. VERSION APRIL 8, 1971 **********''
100PROC LOOK(WORDS,STEMS,CONTENTS,COUNTS=NOTAB,QUERST,NULQRY)$
110 ITEM FAKEH H 6 $
120TABLE WORDS V 100 S 3 $
130 BEGIN
140  ITEM WRDENT H 18 0 0 $
150 END
160
170TABLE STEMS V 100 S 2 $
180 BEGIN
190  ITEM STMENT H 12 0 0 $
200 END
210
220TABLE NOTAB V 100 S 3 $
230 BEGIN
240  ITEM NNOT H 18 0 0 $
250 END
260
270TABLE QUERST V 100 S 6 $
280 BEGIN
290  ITEM QURENT H 12 0 0 $
300  ITEM QUPAIRS H 24 2 0 $
310  ITEM BLOB H 36 0 0 $
320 END
330
340TABLE CONTENTS V 500 S 2 $
350 BEGIN
360  ITEM CONT H 12 0 0 $
520 END
530
540TABLE BUCKET V 10 S 6 $
550 BEGIN
560  ITEM BIGBLOB H 36 0 0 $
570  ITEM STM H 12 0 0 $
580 END
590 ARRAY FAKEH 241 I 35 U $ OVERLAY FAKEH=BUCKET $
600ITEM NULQRY S V(OFF) V(ON) $
610ITEM LESSQ S V(LESS) V(EQL) V(GRTR) $
620ITEM HERE I 36 S $
630ITEM LASTBUCK I 36 S P 0 $''LAST BUCKET IN CORE POINTER''
640ITEM LAST B $''"TRUE" ONLY IF STEM MATCHES LAST ITEM IN BUCKET''
650ITEM THERE I 36 S $
660ITEM NULLHEAD I 36 S $''HEAD OF "NULLS" LIST''
670TABLE COUNTS V 100 S 1 $
671 BEGIN
672  ITEM COUNTER I 36 S 0 0 $
673 END
674ITEM LINEND H 6 $
```

Figure VI.24. LOOK Listing

```
675ITEM LINENDF I 36 S P Ø(770120202020) $
676ØVERLAY LINEND=LINENDF$
680TABLE NULLS V 100 S 1 $
690  BEGIN
700   ITEM NULLA I 36 S 0 0 $
710  END
720
730''     EXPLANATIØN ØF THE METHØD ØF "LØØKUP"
740       EACH ITEM IN THE "STEMS" TABLE IS FIRST
750       LØØKED UP IN TTE "CØNTENTS" TABLE TØ
760       DETERMINE WHICH "BUCKET" ØF DISC TØ USE.
770       "HERE" WILL PØINT TØ THE CØRRECT DISC
780       BUCKET.  IF THE INCØMING STEM MATCHES ØNE
790       ØF THE ITEMS IN "CØNTENTS", WE DØ NØT HAVE
800       TØ SEARCH THE DISC BUCKET, AS THE "CØNTENTS"
810       TABLE CØNSISTS ØF THE LAST STEMS IN EACH ØF
820       THE 500 DISC BUCKETS.  ELSE, THE INCØMING
830       STEM IS BINARY SEARCHED IN THE CØRRECT BUCKET.
840
850       UPØN FINDING A STEM IN THE DISC BUCKET, THE STEM
860       AND ITS CØNCEPT/WEIGHT PAIRS ARE CØPIED INTØ THE
870       "QUERST" TABLE FØR SUBSEQUENT ANALYSIS BY THE
880       "QCVBLD" RØUTINE FØR RETRIEVAL BUILDING        ''
890
900BEGIN ''LØØK'' ØPENDI$ ''ØPEN DICTIØNARY FILE ''
905NULQRY=V(ØFF)$ ''INITIAL STATE = V(ØFF) ''
910LASTBUCK=150$''AN IMPØSSIBLY HIGH NUMBER ''
920''SET RELEVANT NENTS''
930NENT(NØTAB)=0$
940NENT(QUERST)=0$
960NENT(BUCKET)=10$''DISC "BUCKET" SIZE''
970''CHECK FØR "NØ-STEMS-TØ-PRØCESS" CØNDITIØN''
980IF NENT(STEMS) EQ 0 $
990 BEGIN
1000  NULQRY=V(ØN)$''RAISE THE FLAG''
1010  GØTØ BAD $
1020 END
1030''ACTUAL EXECUTIØN STARTS HERE''
1040FØR S=ALL(STEMS)$
1050BEGIN
1060''PRØCESS THE ENTIRE CØNTENTS ØF THE "STEMS" TABLE''
1070''FIRST LØØK UP IN "TABLE-ØF-CØNTENTS" TØ FIND BUCKET #''
1080 HUNT2A(STMENT($S$),CØNTENTS,0,NENT(CØNTENTS)-1=
1090          HERE,LESSQ)$
1100 IFEITH LESSQ EQ V(EQL) $''MATCHES LAST BUCKET ITEM''
1110  BEGIN ''EQUAL PRØCESSING''
1120   LAST=TRUE$
1130   GØTØ PULL$
1140  END
1150 ØRIF LESSQ EQ V(LESS)$''IN THE RANGE ØF THIS BUCKET''
1160  GØTØ PULL$
1170 END ''THIS "IFEITH"''
```

Figure VI.24. (Continued)

```
1180''MUST BE ØNE GREATER THAN THIS BUCKET, BY THE BINARY
1190  SEARCH ALGØRITHM METHØD''
1200HERE=HERE+1$
1210IF HERE GR NENT(CØNTENTS)$''HIGHER THAN LAST ENTRY IN DICT.''
1220  GØTØ SØRRY$
1230PULL. ''PREPARE TØ FETCH (IF NECESSARY) THE BUCKET ØF ENTRIES''
1240IF HERE NQ LASTBUCK$''IS BUCKET ALREADY IN CØRE''
1250 BEGIN
1260   RDBUCK(HERE+1=BUCKET)$ ''FETCH STUFF FRØM DISC''
1270   LASTBUCK=HERE$
1280   END
1290IF LAST$''MATCHES LAST ITEM IN BUCKET''
1300 BEGIN
1310  THERE=9$''LAST ITEM IN BUCKET''
1320  GØTØ PUT$
1330 END
1340''ELSE, TRY TØ FIND "STMENT(S)" IN CURRENT BUCKET''
1350HUNT2B(STMENT($S$),BUCKET,0,NENT(BUCKET)-1=THERE)$
1360IF THERE LS 0 $
1370 GØTØ SØRRY$''NØT FØUND IN STEM DICTIØNARY''
1380PUT.
1390BLØB($NENT(QUERST)$)=BIGBLØB($THERE$) $
1400NENT(QUERST)=NENT(QUERST)+1 $
1410TEST S$
1420SØRRY. ''NØT A DICTIØNARY STEE -- TABLE APPRØPRIATE ACTIØN''
1430NNØT($NENT(NØTAB)$)=WRDENT($S$) $
1431BYTE($CØUNTER($S$),2$)(NNØT($NENT(NØTAB)$))=
1432BYTE($0,2$)(LINEND)$
1440NENT(NØTAB)=NENT(NØTAB)+1$
1450WRDENT($S$)=18H(                  )$
1460STMENT($S$)=12H(            )$
1461CØUNTER($S$)=0$
1470TEST S $
1480END''ALL STEMS CHECK, "S" NØ LØNGER DEFINED''
1490''    BEGIN FINAL CLEANUP ØF TABALES FØR ØUTPUT''
1500''SEE IF ANY STEMS WERE REJECTED''
1510IF NENT(NØTAB) EQ 0$
1520 BEGIN ''ALL STEMS WERE VALID''
1530  NULQRY=V(ØFF)$''LØWER FLAG''
1540  CLØSDI$ RETURN $
1550 END
1560BLARB. ''SEND USER "EXCLUDED LIST"''
1570NULLHEAD=0$
1580NENT(NULLS)=0$
1590FØR W=ALL(WØRDS)$ ''APPLIES TØ "STEMS" TØØ ''
1600BEGIN
1610 IF WRDENT($W$) EQ
1620 18H(                  )$
1640  BEGIN
1650    NULLA($NULLHEAD+NENT(NULLS)$)=W$ ''ENTER PØINTER TØ NULL''
1660    NENT(NULLS)=NENT(NULLS)+1$
1670    TEST W $
```

Figure VI.24. (Continued)

```
1680  END
1690 IF NENT(NULLS) EQ 0$
1700   TEST W $
1710'' ELSE, MOVE LAST GOOD ENTRIES TO LEADING CANDIDATE
1720  IN THE "NULLS" QUEUE.''
1730 WRDENT($NULLA($NULLHEAD$)$)=WRDENT($W$) $
1740 WRDENT($W$)=18H(                  )$
1750 STMENT($NULLA($NULLHEAD$)$)=STMENT($W$) $
1760 STMENT($W$)=12H(            )$
1761 NULLA($NULLHEAD+NENT(NULLS)$)=W $''POINTER TO "NULL" ''
1763COUNTER($NULLA($NULLHEAD$)$)=COUNTER($W$) $
1764COUNTER($W$)=0$''DON'T FORGET TO ZERO THE COUNTER''
1770 NULLHEAD=NULLHEAD+1$ ''LOGICALLY DELETE HEAD OF QUEUE''
1790END
1800''WHEN COMPRESSION IS DONE, CONTINUE HERE.''
1810NENT(WORDS)=NENT(WORDS)-NENT(NOTAB)$
1820NENT(STEMS)=NENT(WORDS) $
1830IF NENT(WORDS) LQ 0 $
1840 NULQRY=V(ON)$''RAISE THE "BAD" FLAG''
1850''TELL THE USER THAT HE EMPLOYED SOME NON-DICT. WORDS''
1860MESAGE(10)$
1870FOR B=ALL(NOTAB)$''SEND LIST OF "BAD" WORDS TO THE USER''
1880 SEND(NNOT($B$),3,18) $
1890IF NULQRY EQ V(ON)$''WAS THE FLAG "RAISED" ''
1900BAD. MESAGE(11)$ ''TELL HIM''
1910CLOSDI$ RETURN$
1920END ''OF L O O K  UP PROCEDURE''
```

Figure VI.24. (Concluded)

contains 500 end-of-bucket indicators, instead of the planned 125. It is to be noted that the call from LOOK raises the block number by one before invoking RDBUCK, because the CONTENTS table is addressed according to the JOVIAL convention: i.e., from 0 (zero) to NENT (CONTENTS) - 1. It is to be noted that two other entry points are provided: OPENDI and CLOSDI to open and close respectively the permfile on which the dictionary is stored.

VI. 2. 3. 5 RETRIV

This overlay performs the actual document retrieval process, including accessing of the inverted lists for concepts, computing the correlation between the documents and the query, and sorting both the correlation tables and TEMP, once the new entries have been entered. The following text contains the flowchart comments keyed to Figure VI. 25.

1. Preset item QRYMAG, the magnitude of the QUERY vector, to 0.0. Preset item HITS, the number of documents retrieved by the current query, to 0. Set NENT(ORTAB), the table of document ascession numbers and the QUERY-DOCUMENT correlations, to 0.

2. Zero fill arrayFLAGS, the array which tallies the documents related to the input QUERY; zero-fill CORTAB, the QUERY-DOCUMENT correlation table.

3. Set the "last retrieval" flag in the document table DOCMNT, to zero ("off") for all previously retrieved documents found in that table.

4. Send the message, "***ANALYZING QUERY***," to the user.

5. Sequentially fetch each entry of the sparse QUERY VECTOR.

6. Is the value of the current QUERY VECTOR entry greater than 0.0? (Concept number is implicit by position).

Yes => 7 No => 13

7. Add to the growing QRYMAG the square of the correlation weight of the concept currently isolated from the sparse QUERY VECTOR.

8. Call subroutine FETINO, the procedure which given a concept number, return a list of all documents containing that concept. The concept number is determined by the current value of the index of the loop which streams through the sparse Query Vector (recall that Concept # is implicit by position); the buffer on which the document # list is stored is array INVERT.

9. Sequentially fetch each entry of the inverted list buffer, INVERT.

10. Is the value of the current entry of the inverted list equal to 0 (signifying END-OF-LIST)?

Yes => 12 No => 11

11. Flag bit 'n' in the FLAGS array, where 'n' is the document number.

12. End-of-loop which searches the inverted list. Loop completed?

Yes => 13 No => 9

13. End-of-loop which streams through the QUERY vector. Loop completed?

Yes => 14 No => 5

14. Send two successive message to the user, via subroutine SEND:

1. "*** QUERY ANALYSIS COMPLETED ***"
2. "*** QUERY-DOCUMENT CORRELATION BEGUN ***"

15. Set item SHIFTS, a counter which tallies the relative bit position in a word isolated from a array FLAGS, to 1.

16. Sequentially fetch each entry of the document number flag list, array FLAGS.

17. Is the current word isolated from array FLAGS EQ 0?

Yes => 30 No => 18

(In order to gain speed the entire word is tested for a value of 0, instead of testing all 36 bits successively. If indeed any bits are set to 1, then each bit is sequentially tested).

18. Transfer if bit 0 is "ON".

Bit 0 "ON"?

Yes => 23   No => 19

19. Shift the word 1 bit to the left (vacated position at bit 35 is zero-filled). Increment counter SHIFTS.

20. Bit 0 "ON"?

Yes => 23   No => 21

21. If all "ON" bits of the current word from FLAGS have been encountered and shifted, the word will now contain all zeros.

Are the contents of the current work equal to 0?

Yes => 22   No => 19

22. 1 => SHIFTS.

23. Store the word isolated from FLAGS in a temporary storage cell, item TEMP.

24. Calculate the document retrieval number, item RETNUM.

25. Is this document in the list of documents which the user has specified to be excluded from the retrieval?

Yes => 29   No => 26

26. Call subroutine FETDCV which, given a document ascession number, RETNUM, returns the proper Document Concept Vector in buffer DCVBUF.

27. Calculate the cosine between the QUERY VECTOR and the DOCUMENT CONCEPT VECTOR.

28. Store the document ascession number, RETNUM, and the cosine correlation value, TOTMAG, in table CORTAB.

29. C(TEMP) => C(Q).

30. End of the loop which streams through array FLAGS and calculates the QUERY-DOCUMENT CORRELATIONS.

Loop completed?

Yes => 31   No => 16

31. Sort the correlation table, CORTAB, using a D. L. Shell sinking-sort, by descending correlation.

32. Enter the data for the 64 (or fewer) documents which correlated most strongly with the query in the table DOCMNT.

33. Return to the calling routine, the DIALOGUE PROCESSOR.

VI. 2. 3. 6 NUMBER Procedure

At various points in the man-machine dialogue the user is called upon to specify document numbers. Overlay procedure NUMBER interacts with the user to obtain a document number or a series of numbers. Any one of three types of specification may be used:

1) a single temporary I. D. number or a range of numbers
2) a single accession number or range of numbers
3) the whole TEMP table (all documents retrieved)

The following is a BNF description of the syntax of the permissible number input

1. <DOCID> ::= <LSIDE> | <LSIDE><DELIM> <RSIDE> TEMP
2. <RSIDE> ::= <DIGITS>
3. <SLIDE> ::= <DIGITS> | <DEF> <DIGITS>
4. <DEF> ::= A <BLANKS> | T <BLANKS>
5. <DELIM> ::= <BLANKS> - <BLANKS>
6. <BLANKS> ::= zero or more space characters
7. <DIGITS> ::= one to five numerals, 0-9

Violations of the above syntax will result in error recovery within NUMBER. The following are examples of input records:

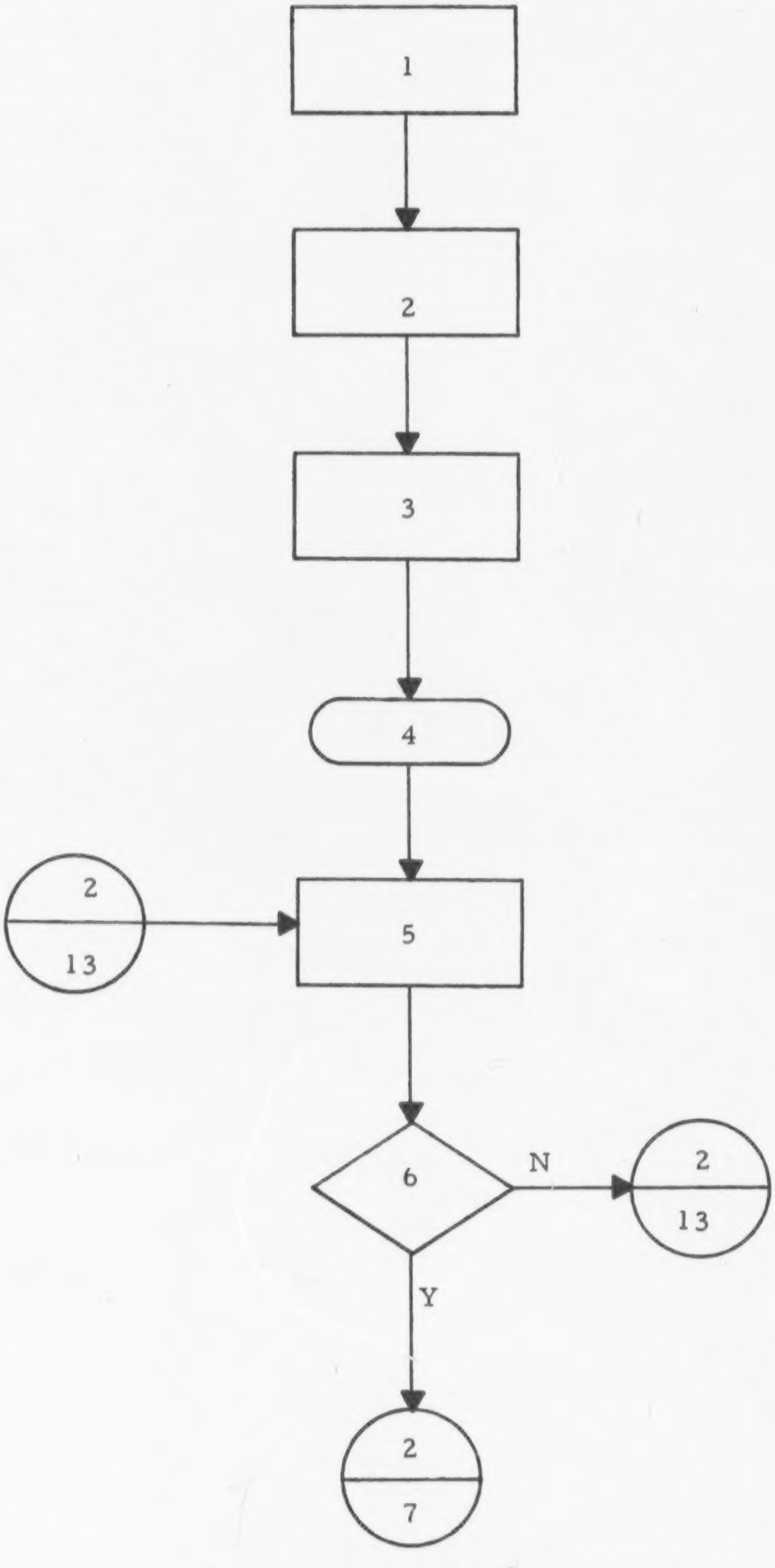

1
2
3
4
2
13
5
6
N
2
13
Y
2
7

Figure VI. 25. RETRIV Flowchart

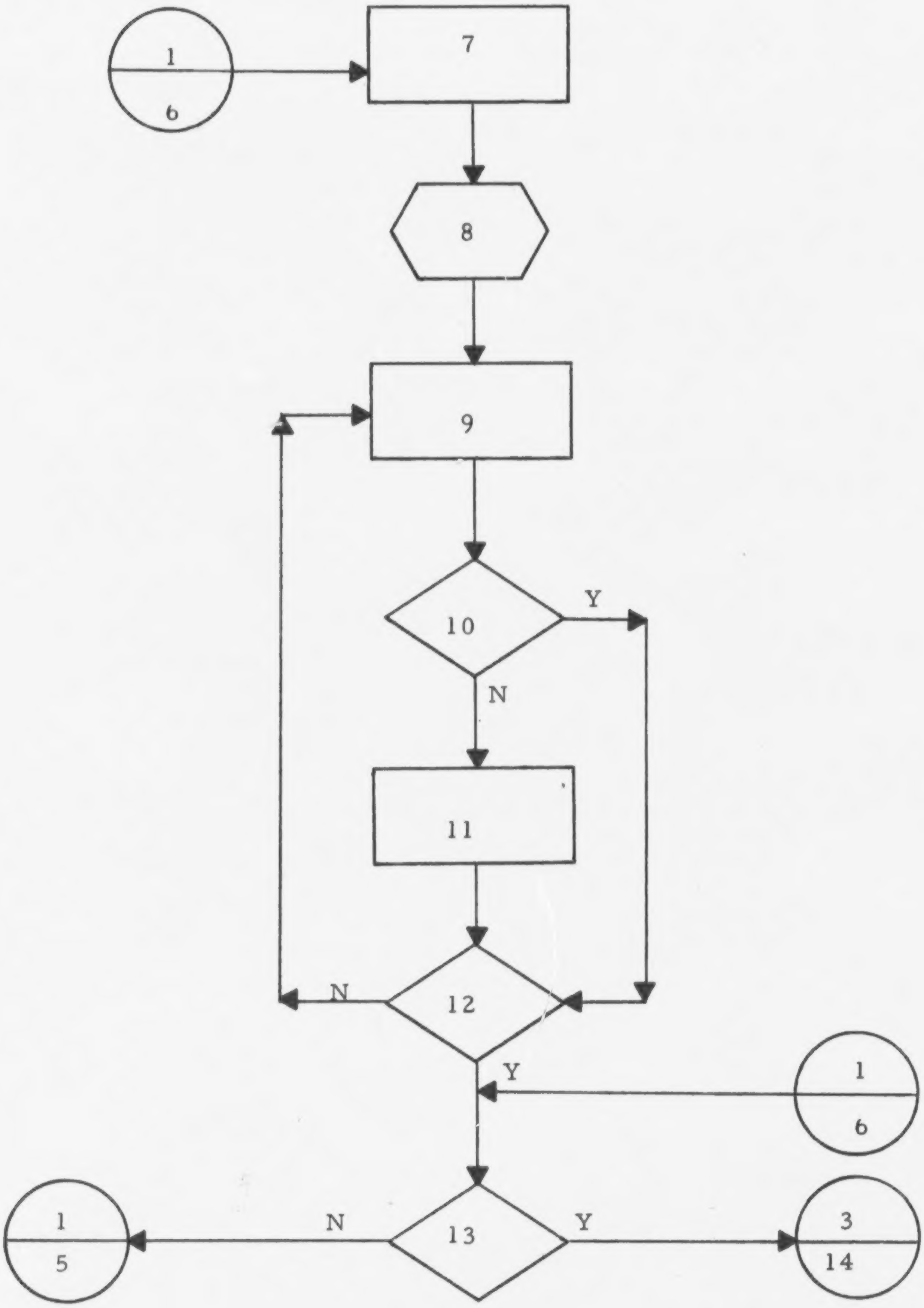

1
6
7
8
9
10
Y
N
11
N
12
Y
1
6
1
5
N
13
Y
3
14

Figure VI. 25. (Continued)

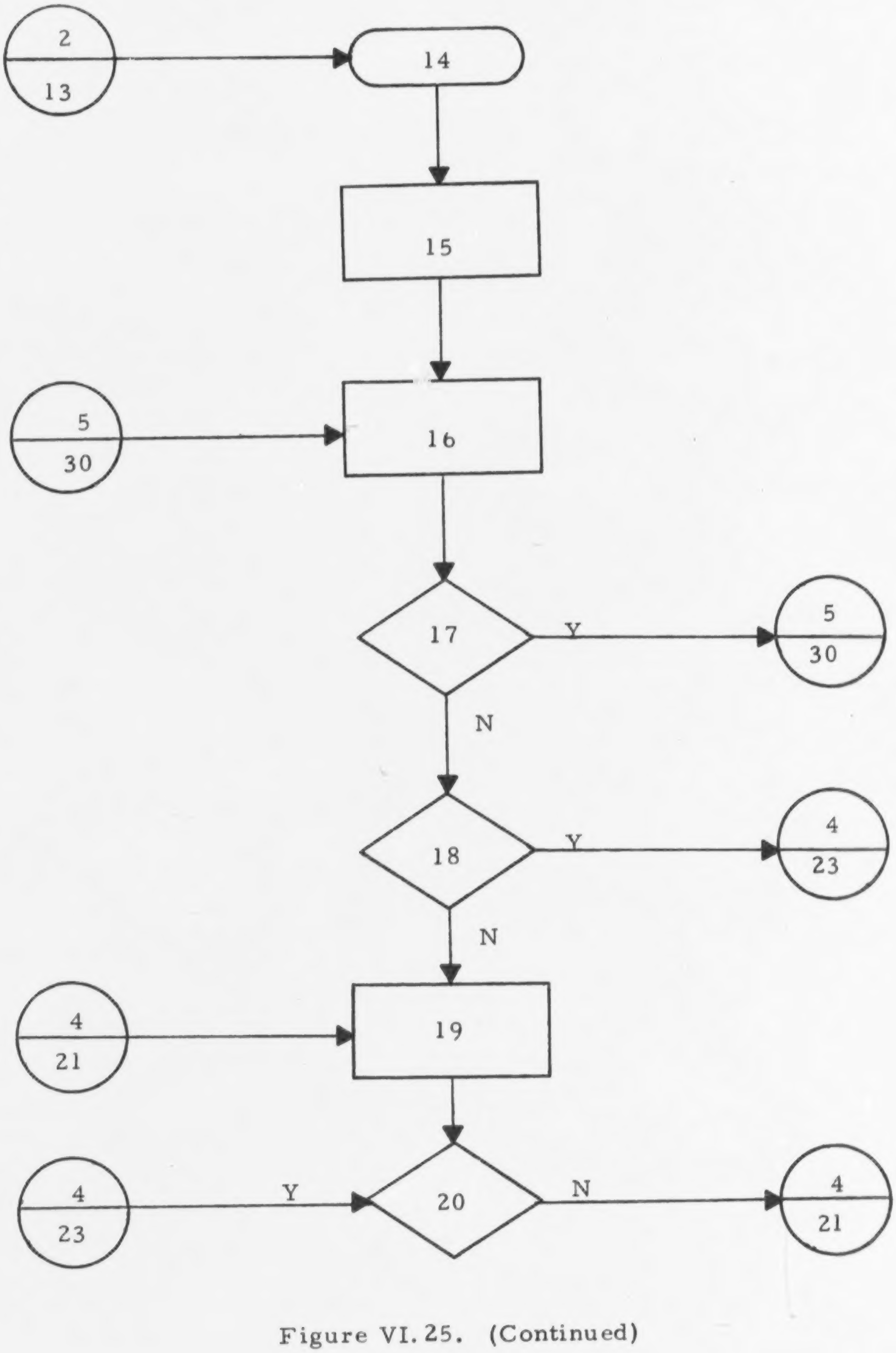
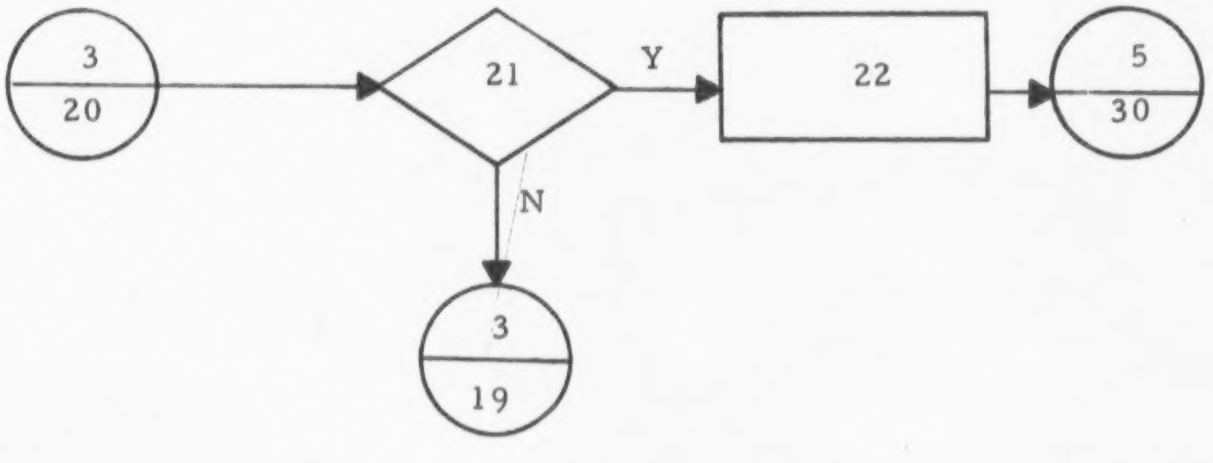

2
13
14
15
5
30
16
17
Y
5
30
N
18
Y
4
23
N
4
21
19
4
23
Y
20
N
4
21

Figure VI. 25. (Continued)


3
20
21
Y
22
5
30
N
3
19

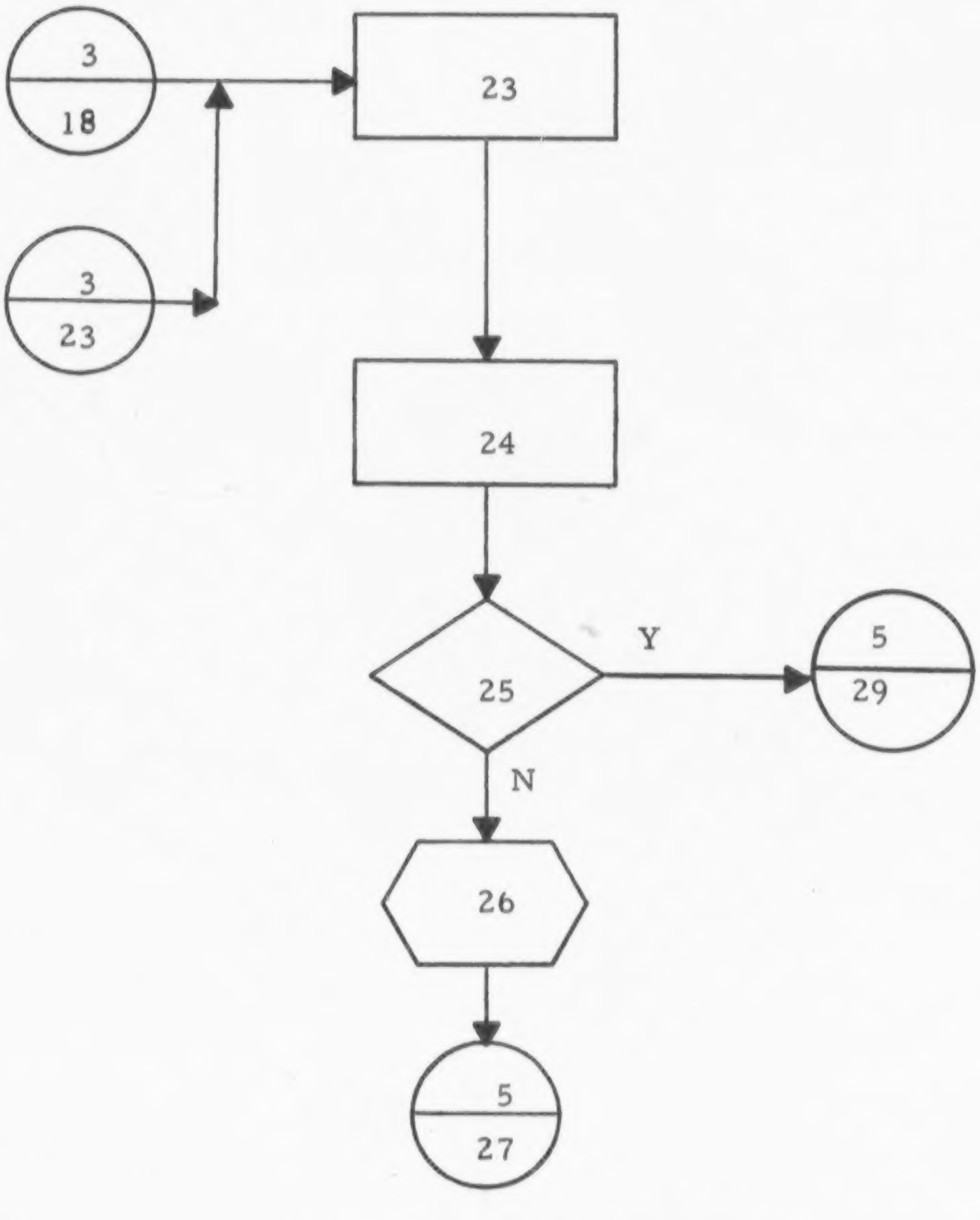

3
18
23
3
23
24
25
Y
5
29
N
26
5
27

Figure VI. 25. (Continued)

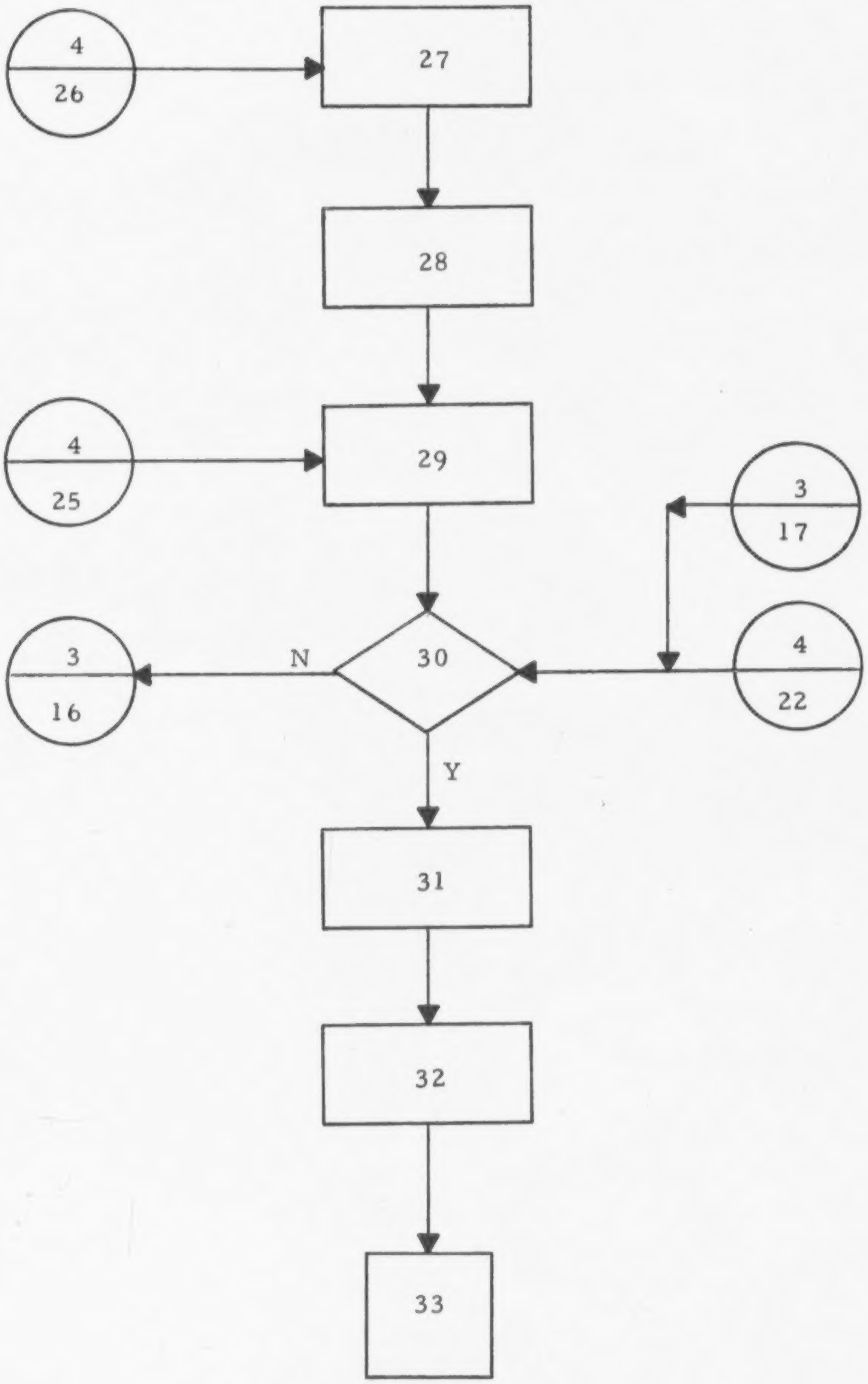
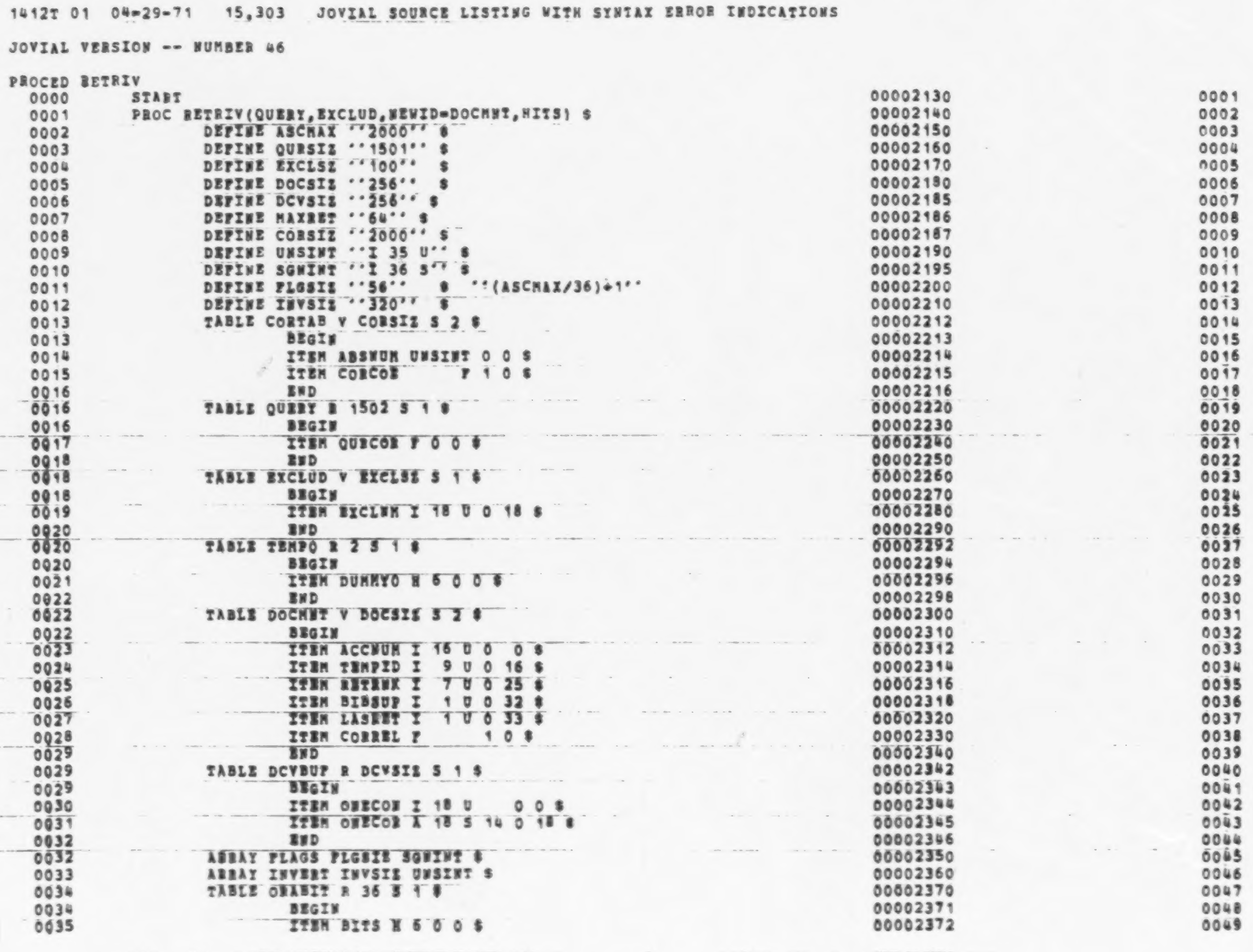

4
26
27
28
4
25
29
3
17
3
16
N
30
4
22
Y
31
32
33

Figure VI. 25. (Concluded)



```
1412T 01  04-29-71   15,303   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROCED RETRIV
  0000      START                                                                   00002130                        0001
  0001      PROC RETRIV(QUERY,EXCLUD,NEWID=DOCMNT,HITS) $                           00002140                        0002
  0002             DEFINE ASCMAX ''2000'' $                                         00002150                        0003
  0003             DEFINE QURSIZ ''1501'' $                                         00002160                        0004
  0004             DEFINE EXCLSZ ''100''  $                                         00002170                        0005
  0005             DEFINE DOCSIZ ''256''  $                                         00002180                        0006
  0006             DEFINE DCVSIZ ''256'' $                                          00002185                        0007
  0007             DEFINE MAXRET ''64'' $                                           00002186                        0008
  0008             DEFINE CORSIZ ''2000'' $                                         00002187                        0009
  0009             DEFINE UNSINT ''I 35 U'' $                                       00002190                        0010
  0010             DEFINE SGNINT ''I 36 S'' $                                       00002195                        0011
  0011             DEFINE FLGSIZ ''56''   $  ''(ASCMAX/36)+1''                      00002200                        0012
  0012             DEFINE INVSIZ ''320''  $                                         00002210                        0013
  0013             TABLE CORTAB V CORSIZ S 2 $                                      00002212                        0014
  0013                     BEGIN                                                    00002213                        0015
  0014                     ITEM ABSNUM UNSINT 0 0 $                                 00002214                        0016
  0015                     ITEM CORCOR        F 1 0 $                               00002215                        0017
  0016                     END                                                      00002216                        0018
  0016             TABLE QUERY R 1502 S 1 $                                         00002220                        0019
  0016                     BEGIN                                                    00002230                        0020
  0017                     ITEM QURCOR F 0 0 $                                      00002240                        0021
  0018                     END                                                      00002250                        0022
  0018             TABLE EXCLUD V EXCLSZ S 1 $                                      00002260                        0023
  0018                     BEGIN                                                    00002270                        0024
  0019                     ITEM EXCLNM I 18 U 0 18 $                                00002280                        0025
  0020                     END                                                      00002290                        0026
  0020             TABLE TEMPO R 2 S 1 $                                            00002292                        0027
  0020                     BEGIN                                                    00002294                        0028
  0021                     ITEM DUMMYO H 6 0 0 $                                    00002296                        0029
  0022                     END                                                      00002298                        0030
  0022             TABLE DOCMNT V DOCSIZ S 2 $                                      00002300                        0031
  0022                     BEGIN                                                    00002310                        0032
  0023                     ITEM ACCNUM I 16 U 0  0 $                                00002312                        0033
  0024                     ITEM TEMPID I  9 U 0 16 $                                00002314                        0034
  0025                     ITEM RETRNK I  7 U 0 25 $                                00002316                        0035
  0026                     ITEM BIBSUP I  1 U 0 32 $                                00002318                        0036
  0027                     ITEM LASRET I  1 U 0 33 $                                00002320                        0037
  0028                     ITEM CORREL F      1 0 $                                 00002330                        0038
  0029                     END                                                      00002340                        0039
  0029             TABLE DCVBUF R DCVSIZ S 1 $                                      00002342                        0040
  0029                     BEGIN                                                    00002343                        0041
  0030                     ITEM ONECON I 18 U    0 0 $                              00002344                        0042
  0031                     ITEM ONECOR A 18 S 14 0 18 $                             00002345                        0043
  0032                     END                                                      00002346                        0044
  0032             ARRAY FLAGS FLGSIZ SGNINT $                                      00002350                        0045
  0033             ARRAY INVERT INVSIZ UNSINT $                                     00002360                        0046
  0034             TABLE ONEBIT R 36 S 1 $                                          00002370                        0047
  0034                     BEGIN                                                    00002371                        0048
  0035                     ITEM BITS H 6 0 0 $                                      00002372                        0049
```

Figure VI. 25a. RETRIV Listing

```
1412T 01  04-29-71   15,303    JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0035                                    BEGIN                                  00002374                       0050
  0035                                    O(400000000000)                        00002376                       0051
  0035                                    O(200000000000)                        00002378                       0052
  0035                                    O(100000000000)                        00002380                       0053
  0035                                    O(040000000000)                        00002382                       0054
  0035                                    O(020000000000)                        00002384                       0055
  0035                                    O(010000000000)                        00002386                       0056
  0035                                    O(004000000000)                        00002388                       0057
  0035                                    O(002000000000)                        00002390                       0058
  0035                                    O(001000000000)                        00002392                       0059
  0035                                    O(000400000000)                        00002394                       0060
  0035                                    O(000200000000)                        00002396                       0061
  0035                                    O(000100000000)                        00002398                       0062
  0035                                    O(000040000000)                        00002400                       0063
  0035                                    O(000020000000)                        00002402                       0064
  0035                                    O(000010000000)                        00002404                       0065
  0035                                    O(000004000000)                        00002406                       0066
  0035                                    O(000002000000)                        00002408                       0067
  0035                                    O(000001000000)                        00002410                       0068
  0035                                    O(000000400000)                        00002412                       0069
  0035                                    O(000000200000)                        00002414                       0070
  0035                                    O(000000100000)                        00002416                       0071
  0035                                    O(000000040000)                        00002418                       0072
  0035                                    O(000000020000)                        00002420                       0073
  0035                                    O(000000010000)                        00002422                       0074
  0035                                    O(000000004000)                        00002424                       0075
  0035                                    O(000000002000)                        00002426                       0076
  0035                                    O(000000001000)                        00002428                       0077
  0035                                    O(000000000400)                        00002430                       0078
  0035                                    O(000000000200)                        00002432                       0079
  0035                                    O(000000000100)                        00002434                       0080
  0035                                    O(000000000040)                        00002436                       0081
  0035                                    O(000000000020)                        00002438                       0082
  0035                                    O(000000000010)                        00002440                       0083
  0035                                    O(000000000004)                        00002442                       0084
  0035                                    O(000000000002)                        00002444                       0085
  0035                                    O(000000000001)                        00002446                       0086
  0035                                    END                                    00002450                       0087
  0036                       END                                                 00002510                       0088
  0036             ITEM NEWID  UNSINT $                                          00002513                       0089
  0037             ITEM HITS UNSINT $                                            00002515                       0090
  0038             ITEM     II UNSINT P O $                                      00002516                       0091
  0039             ITEM     JJ UNSINT P O $                                      00002517                       0092
  0040             ITEM     KK UNSINT P O $                                      00002518                       0093
  0041             ITEM     LL UNSINT P O $                                      00002519                       0094
  0042             ITEM SHIFTS UNSINT P O $                                      00002520                       0095
  0043             ITEM RETNUM UNSINT P O $                                      00002530                       0096
  0044             ITEM TEMP   SGNINT P O $                                      00002540                       0097
  0045             ITEM TMPCON UNSINT P O $                                      00002541                       0098
  0046             ITEM TMPCOR F $                                               00002542                       0099
```

Figure VI. 25a. (Continued)

```
1412T 01  04-29-71   15,303   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0047             ITEM TMPQRY F $                                  00002543            0100
 0048             ITEM DOTMAG F $                                  00002544            0101
 0049             ITEM TOTMAG F $                                  00002545            0102
 0050             ITEM QRYMAG F $                                  00002550            0103
 0051             ITEM DCVMAG F $                                  00002560            0104
 0052       BEGIN ''RETRIV''                                       00003500            0105
 0053             QRYMAG=0.0 $                                     00003505            0106
 0054             HITS=0 $                                         00003510            0107
 0055             NENT(CORTAB)=0 $                                 00003512            0108
 0056             FOR I=0,1,FLGSIZ-1 $                             00003514            0109
 0057                     BEGIN                                    00003516            0110
 0058                     FLAGS($I$)=0 $                           00003518            0111
 0059                     END                                      00003520            0112
 0060             FOR I=0,1,CORSIZ-1 $                             00003522            0113
 0061                     BEGIN                                    00003524            0114
 0062                     ABSNUM($I$)=0 $                          00003526            0115
 0063                     CORCOR($I$)=0.0 $                        00003528            0116
 0064                     END                                      00003530            0117
 0065             FOR I=0,1,NENT(DOCMNT)-1 $                       00003532            0118
 0066                     BEGIN                                    00003534            0119
 0067                     LASRET($I$)=0 $                          00003536            0120
 0068                     END                                      00003538            0121
 0069             SEND(24H(***ANALYZING QUERY***   ),4,24) $       00003539            0122
 0070             FOR I=1,1,QURSIZ $                               00003540            0123
 0071                     BEGIN                                    00003542            0124
 0072                     IF QURCOR($I$) EQ 0.0 $                  00003544            0125
 0073                     TEST I $                                 00003546            0126
 0074       NITCON.                                                00003590            0127
 0074                     QRYMAG=QRYMAG+(QURCOR($I$)*QURCOR($I$)) $ 00003595           0128
 0075                     FETINV(I,INVERT) $                       00003600            0129
 0076                     FOR J=1,1,INVSIZ $                       00003610            0130
 0077                               BEGIN                          00003620            0131
 0078                               DIRECT                         00003630            0132
 0079             LDA     J                                        00003640            0133
 0080             LDQ     INVERT,AL                                00003650            0134
 0081             TNZ     SETBIT                                   00003660            0135
 0082                               JOVIAL                         00003670            0136
 0083                               TEST I $                       00003680            0137
 0084       SETBIT.                                                00003690            0138
 0084                               DIRECT                         00003700            0139
 0085             SBQ     1,DL                                     00003702            0140
 0086             TZE     NUMONE                                   00003704            0141
 0087             DIV     36,DL                                    00003710            0142
 0088             LDA     ORABIT+1,AL                              00003720            0143
 0089             ORSA    FLAGS,QL                                 00003730            0144
 0090                               JOVIAL                         00003740            0145
 0091                               TEST J $                       00003742            0146
 0092                               DIRECT                         00003746            0147
 0093       NUMONE LDA    ORABIT+1                                 00003748            0148
 0094             ORSA    FLAGS                                    00003750            0149
```

Figure VI.25a. (Continued)

```
1412T 01  04-29-71   15,303   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0095                            JOVIAL                                          00003752            0150
 0096                            END                                             00003754            0151
 0097                      END                                                   00003780            0152
 0098             SEND(30H(***QUERY ANALYSIS COMPLETED***),5,30) $               00003785            0153
 0099             SEND(42H(***QUERY-DOCUMENT CORRELATION BEGUN***    ),7,42)$    00003786            0154
 0100             SHIFTS=1 $                                                     00003800            0155
 0101             FOR I=0,1,FLGSIZ-1 $                                           00003810            0156
 0102                      BEGIN                                                 00003820            0157
 0103                      DIRECT                                                00003828            0158
 0104             LDA      I                                                     00003830            0159
 0105             LDQ      FLAGS,AL                                              00003840            0160
 0106             TNZ      HITDOC                                                00003850            0161
 0107                      JOVIAL                                                00003860            0162
 0108                      TEST I $                                              00003870            0163
 0109                      DIRECT                                                00003880            0164
 0110      HITDOC TMI      GOTCHA                                                00003900            0165
 0111      SPINIT AOS      SHIFTS                                                00003910            0166
 0112             QLS      1                                                     00003920            0167
 0113             TMI      GOTCHA                                                00003930            0168
 0114             CMPQ     0,DL                                                  00003940            0169
 0115             TNZ      SPINIT                                                00003950            0170
 0116                      JOVIAL                                                00003970            0171
 0117                      SHIFTS=1 $                                            00003975            0172
 0118                      TEST I $                                              00003980            0173
 0119      GOTCHA,                                                               00003990            0174
 0119                      DIRECT                                                00004000            0175
 0120             STQ      TEMP                                                  00004010            0176
 0121                      JOVIAL                                                00004020            0177
 0122                      RETNUM=(I*36)+SHIFTS $                                00004030            0178
 0123                      FOR J=0,1,NENT(EXCLUD)-1 $                            00004050            0179
 0124                                BEGIN                                       00004060            0180
 0125                                IF EXCLNM($J$) EQ RETNUM $                  00004090            0181
 0126                                GOTO RESET $                                00004100            0182
 0127                                END                                         00004110            0183
 0128      GRABIT,                                                               00005120            0184
 0128                      FETDCV(RETNUM,DCVBUF) $                               00005130            0185
 0129                      DCVMAG=0,0 $                                          00005140            0186
 0130                      DOTMAG=0,0 $                                          00005150            0187
 0131                      TOTMAG=0,0 $                                          00005160            0188
 0132                      FOR J=0,1,DCVSIZ-2 $                                  00005170            0189
 0133                                BEGIN                                       00005180            0190
 0134                                DIRECT                                      00005190            0191
 0135             LDA      J                                                     00005200            0192
 0136             LD~      DCVBUF+1,AL                                           00005210            0193
 0137             TZE      CALCOR                                                00005220            0194
 0138                                JOVIAL                                      00005230            0195
 0139                                TMPCON=ONECON($J$) $                        00005240            0196
 0140                                TMPCOR=ONECOR($J$) $                        00005250            0197
 0141                                TMPQRY=QURCOR($TMPCON$) $                   00005260            0198
 0142                                DCVMAG=DCVMAG+(TMPCOR*TMPCOR) $             00005270            0199
```

Figure VI. 25a. (Continued)

```
1412T 01  04-29-71   15,303   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0143                            IF TMPQRY EQ 0.0 $                          00005280                 0200
 0144                            TEST J $                                    00005290                 0201
 0145                            DOTMAG=DOTMAG+(TMPCOR*TMPCRY) $             00005300                 0202
 0146                            END                                         00005305                 0203
 0147       CALCOR.                                                          00005306                 0204
 0147                   IF DOTMAG EQ 0.0 $                                   00005308                 0205
 0148                   GOTO RESET $                                         00005310                 0206
 0149                   IF DCVMAG EQ 0.0 $                                   00005312                 0207
 0150                   GOTO RESET $                                         00005314                 0208
 0151                   IF QRYMAG EQ 0.0 $                                   00005316                 0209
 0152                   GOTO RESET $                                         00005318                 0210
 0153                   TOTMAG=DOTMAG/((DCVMAG**.5)*(QRYMAG**.5)) $          00005320                 0211
 0154                   IF NENT(CORTAB) LS CORSIZ $                          00005330                 0212
 0155                            BEGIN                                       00005340                 0213
 0156                            ABSNUM($NENT(CORTAB)$)=RETNUM $             00005350                 0214
 0157                            CORCOE($NENT(CORTAB)$)=TOTMAG $             00005360                 0215
 0158                            NENT(CORTAB)=NENT(CORTAB)+1 $               00005370                 0216
 0159                            END                                         00005380                 0217
 0160       RESET.                                                           00005390                 0218
 0160                   DIRECT                                               00005400                 0219
 0161           LDQ     TEMP                                                 00005410                 0220
 0162           TRA     SPINIT                                               00005420                 0221
 0163                   JOVIAL                                               00005430                 0222
 0164                   END                                                  00005440                 0223
 0165       SRT"TO.                                                          00005450                 0224
 0165           LL=2 $                                                       00005460                 0225
 0166       SRTCT1.                                                          00005470                 0226
 0166           IF LL GQ NENT(CORTAB) $                                      00005480                 0227
 0167           GOTO SRTCT2 $                                                00005490                 0228
 0168           LL=LL*2 $                                                    00005500                 0229
 0169           GOTO SRTCT1 $                                                00005510                 0230
 0170       SRTCT2.                                                          00005520                 0231
 0170           DD=(LL/2)-1 $                                                00005530                 0232
 0171       SRTCT3.                                                          00005540                 0233
 0171           II=0 $                                                       00005550                 0234
 0172           JJ=0 $                                                       00005560                 0235
 0173           KK=0 $                                                       00005570                 0236
 0174           DIRECT                                                       00005580                 0237
 0175       SRTCT4 LDA      II                                               00005590                 0238
 0176           ALS     1                                                    00005600                 0239
 0177           EAX7    CORTAB+1,AL                                          00005610                 0240
 0178           ARS     1                                                    00005620                 0241
 0179           ADA     DD                                                   00005630                 0242
 0180           ALS     1                                                    00005640                 0243
 0181           EAX6    CORTAB+1,AL                                          00005650                 0244
 0182           FLD     1,7                                                  00005660                 0245
 0183           FCMP    1,6                                                  00005670                 0246
 0184           TMI     SRTCT7                                               00005680                 0247
 0185           JOVIAL                                                       00005690                 0248
 0186       SRTCT5.                                                          00005700                 0249
```

Figure VI.25a. (Continued)

```
1412T 01  04-29-71   15,303   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0186          IF II+DD+1 LS NENT(CORTAB) $                               00005710     0250
  0187                 BEGIN                                               00005720     0251
  0188                 DIRECT                                              00005730     0252
  0189          AOS    II                                                  00005740     0253
  0190          TRA    SRTCT4                                              00005750     0254
  0191                 JOVIAL                                              00005760     0255
  0192                 END                                                 00005770     0256
  0193          DD=(DD-1)/2 $                                              00005780     0257
  0194          IF DD LQ 0 $                                               00005790     0258
  0195          GOTO ENDSRT $                                              00005800     0259
  0196          GOTO SRTCT3 $                                              00005810     0260
  0197    SRTCT7.                                                          00005820     0261
  0197          DIRECT                                                     00005830     0262
  0198          LDAQ   0,7                                                 00005840     0263
  0199          STAQ   TEMPO+1                                             00005850     0264
  0200          LDAQ   0,6                                                 00005860     0265
  0201          STAQ   0,7                                                 00005870     0266
  0202          LDAQ   TEMPO+1                                             00005880     0267
  0203          STAQ   0,6                                                 00005890     0268
  0204          JOVIAL                                                     00005900     0269
  0205          JJ=II-DD $                                                 00005910     0270
  0206          IF JJ LS 0 $                                               00005920     0271
  0207          GOTO SRTCT5 $                                              00005930     0272
  0208          KK=II $                                                    00005940     0273
  0209          DIRECT                                                     00005950     0274
  0210    SRTCT8 LDA   JJ                                                  00005960     0275
  0211          ALS    1                                                   00005970     0276
  0212          EAX7   CORTAB+1,AL                                         00005980     0277
  0213          LDA    KK                                                  00005990     0278
  0214          ALS    1                                                   00005995     0279
  0215          EAX6   CORTAB+1,AL                                         00006000     0280
  0216          FLD    1,7                                                 00006010     0281
  0217          FCMP   1,6                                                 00006020     0282
  0218          TZE    SRTCT5                                              00006030     0283
  0219          TPL    SRTCT5                                              00006040     0284
  0220          LDAQ   0,7                                                 00006050     0285
  0221          STAQ   TEMPO+1                                             00006060     0286
  0222          LDAQ   0,6                                                 00006070     0287
  0223          STAQ   0,7                                                 00006080     0288
  0224          LDAQ   TEMPO+1                                             00006090     0289
  0225          STAQ   0,6                                                 00006100     0290
  0226          JOVIAL                                                     00006110     0291
  0227          JJ=JJ-DD $                                                 00006120     0292
  0228          IF JJ LS 0 $                                               00006130     0293
  0229          GOTO SRTCT5 $                                              00006140     0294
  0230          KK=JJ+DD $                                                 00006150     0295
  0231          GOTO SRTCT8 $                                              00006160     0296
  0232    ENDSRT.                                                          00006170     0297
  0232          FOR I=0,1,MAXRET-1 $                                       00008000     0298
  0233                 BEGIN                                               00008010     0299
```

Figure VI. 25a. (Continued)

```
1412T 01  04-29-71   15,303   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0234                       IF NENT(DOCMNT) GQ DOCSIZ $                                  00008020            0300
 0235                       GOTO WRAPUP $                                                00008030            0301
 0236                       IF I EQ NENT(CORTAB) $                                       00008040            0302
 0237                       GOTO WRAPUP $                                                00008050            0303
 0238                       HITS=HITS+1 $                                                00008055            0304
 0239                       ACCNUM(SNENT(DOCMNT)$)=ABSNUM($IS) $                         00008060            0305
 0240                       RETRNK(SNENT(DOCMNT)$)=I+1 $                                 00008070            0306
 0241                       LASRET(SNENT(DOCMNT)$)=1 $                                   00008080            0307
 0242                       CORREL(SNENT(DOCMNT)$)=CORCOE($IS) $                         00008090            0308
 0243                       TEMPID(SNENT(DOCMNT)$)=NEWID $                               00008095            0309
 0244                       NENT(DOCMNT)=NENT(DOCMNT)+1 $                                00008100            0310
 0245                       NEWID=NEWID+1 $                                              00008105            0311
 0246                       END                                                          00008110            0312
 0247       WRAPUP.                                                                      00008120            0313
 0247             RETURN $                                                               00008170            0314
 0248       END                                                                          00008180            0315
 0249       TERM $                                                                       00008190            0316
```

Figure VI. 25a. (Concluded)

input	explanation
A123	Accession number 123
*A123-235	Accession number 123-235 inclusive
T40	Temporary ID number 40
*T40-45	Temporary IDs number 40-45 inclusive
*TEMP	All temporary IDs in the TEMP table

VI. 2. 3. 7 CWPAIR. When modifying the query concept vector directly, the user must specify which concept he wants to modify and the weight he wishes to give to it. Overlay procedure CWPAIR is the interpretor routine for entering these data into the system. Exceptions to the allowed input are recoverable within a limit of four attempts, after which the calling routine is informed of a fatal error. Because direct query modification is deemed an exceptional need, rather than a usual request, the format of data entry is considerably more rigid than the specifications for NUMBER input (c.f., VI. 2. 3. 6). Below is the BNF description for the input:

1. <CWPAIR> :: = <LEFT><DELIM><RIGHT>
2. <LEFT> :: = C <DIGITS> must be exactly four digits
3. <RIGHT> :: = W <DIGITS> | W <S> · <DIGIT> |
   W <DIGIT> · <DIGIT>
4. <S> :: = - negative sign indicator
5. <DIGIT> :: = one number out of the set of numbers 0-9
6. <DIGITS> :: = three and only three numbers out of the set of numbers 0-9
7. <DELIM> :: = , | <BLANKS>
8. <BLANKS> :: = one or more "blank" characters

Leading as well as trailing blank characters are permitted in order to make the C-W pair specification more legible. CWPAIR has automatic error recovery, including a pointer to the offending character on the input page. An upper bound of four attempts is allowed by the routine before reporting failure to the calling program.

---

*Not implemented

VI. 2. 3. 8 OFFline. Preparation of abstract text and bibliographies to be printed offline is a useful function of an on-line system. It was originally planned to include such a function in this version of the Dialogue Processor but, because of other priorities, its implementation has been delayed for the time being. Entry points in the processor code already exist for this function and its addition at a later date would not entail any particularly large effort.

VI. 3 THE STRUCTURE OF THE ON-LINE SYSTEM

This section covers the following topics:

a) Detailed general structure of the system

b) Problems with JOVIAL-compiled procedures and SYMDEF from external routines

c) Methods of assembling the object files of the system; problems encountered and solutions

d) Core allocations, theory and practice of the overlay process as applied to the on-line system

e) Deck set-up for load and execute in one activity; discussion of the temporary version of file loading

f) Saving the main link, file control blocks and all overlays on the H* file; method of re-loading the system-loadable version

In all the discussion outlined above, emphasis is on the detail of the implementation, in contrast to the previous sections of this report which are concerned with the general nature of the project and how to use the software.

VI. 3. 1 General Structure of the System

The Dialogue Processor consists of a main program, externally-compiled service routines, and overlays. The main program itself and all its internal procedures form the system executive, while the externally-compiled routines are needed both by the executive and by the various overlays. During the GELOAD activity, various library routines are also called in from L* to satisfy undefined symbols in all routines; JOVIAL run-time subroutines are also considered in this category. Overlays are allocated to the core area following the main program and all its externally-referenced routines, including those obtained from the system library. Each one of the overlays occupies the same core area and is called in from H* when needed. GECOS-III maintains and positions H* for the system user, thereby relieving him of this particular responsibility.

Owing to the size of the system, a relatively large amount of time is needed to compile all of the JOVIAL and GMAP source-code programs into object code. Therefore, the on-line system is always run from the object decks previously compiled or assembled. In the current implementation it is assumed that all the object decks reside as permanent files in the GECOS-III file system and therefore may be entered in the job stream at run time through the use of $ SELECT control cards; this type of operation is well-illustrated in Section VI. 3. 5 of this report.

At load time the General Loader takes care of linking up the main program and the externally compiled or assembled routines, as well as handling the loading and saving of the overlays. A core map is produced, unless it is specifically inhibited by an option on the $ OPTION card, showing the exact core locations of all parts of the system. As discussed in V. 3. 6, rebuilding the system from object decks is not necessary each time the on-line Dialogue Processor is to be run.

VI. 3. 2 Problems with Global Procedures

During the planning phase of the Dialogue Processor implementation it was hoped that all the global subroutines and procedures could be included in the source code of the main program, allowing the General Loader to perform the necessary linking-up with the overlays. However, the JOVIAL compiler in use at this time on the computing system does not permit the use of the pseudo-operation SYMDEF. The consequence of this fact is the following: either duplicate programs would have to be included in the source code of each overlay which needed a particular routine, or else these routines would have to be compiled in separate activities. Obviously, in view of the self-imposed limitations on core-size, only the latter option was the viable one, least needless duplication take place.

VI. 3. 3 Saving the Independently-Compiled Routines

Because of the problems encountered with SYMDEF usage, as reported above, the decision was made to save each separately-compiled routine as a different object-file within the permanent file system of GECOS-III. During the compile activity, when the DECK option is specified on the JOVIAL compiler control card, an object deck appears on system file C*. The format of the complete deck format is of no concern here, except to note that each such deck begins with a $ OBJECT control card and ends with a $DKEND card, these forming the "left" and "right" brackets around the deck. Unfortunately, system software "bug" prevented the direct saving of this file (C*) during the compile activity, with the result that it was necessary to run a separate BMC (Bulk Media Conversion) to enter the object deck into the permanent file system.

VI. 3. 4 The On-Line System at GELOAD Time

Under GECOS-III each step or phase of a computation session is termed an "activity", within the scope of the whole job. As the Dialogue Processor executes directly from object files, only one activity is necessary to run the system: a GELOAD activity. A GELOAD activity performs two actual functions: linkage-editing (resolving external symbols and linking references together), and core-space allocation, as well as the optional building of the I/O file control blocks, if desired. No copy is kept of the final core image thus developed, unless further control cards are included in the input stream. Performing this function is discussed in the next section.

GELOAD also makes sure that the system library file (L*) is searched during the attempt to resolve undefined external symbols and entry points. In the deck set-up for the one-activity load & go method each of the separately-compiled object decks is a different file; the different files are entered into the input job stream through the use of the $ SELECT system control card. Once all the compiled object files are read in, then a search is made of L* to satisfy the remaining references. Overlays are treated slightly differently.

Overlaid portions of the system are treated after all of the main program, subroutines, and library routines referenced are already loaded and linked. All references to symbols contained in the overlaid areas are still undefined at the point when overlay processing begins. Each separate link is loaded according to the information given on the individual $ LINK cards present in the input job stream. After each link is processed and before the next link is started, a search is once again made of the system library file for any additional unresolved symbols (these might be library routines not used by the main link itself but needed in this particular link). After each link is completely processed, it is in turn written out on file H* in what is called "system-loadable" format. When all links are processed, the main link and file control blocks, if specified, are also written out on H*. If execution is specified (no NOGO option used), control is passed to the main entry point (primary SYMDEF) in the main link.

At the end of all link processing by the General Loader, all references to external symbols should be linked to their corresponding entry points. Should any symbols remain undefined at this point, MME GEBORT instructions will be inserted by the General Loader, causing fatal execution errors if these routines are referenced during program operation.

VI.3.5 Saving the Links Built During the GELOAD Activity and Subsequently Re-Loading Them

Recall that program links are written out on system file H* during the processing of the object files by GELOAD. Because the

GELOAD activity is necessary only once, and because this activity is somewhat time-consuming, it is desirable to save the H* file and re-load it when the system is to be run again. This facility is available through the use of the routine RLHS, a library program. In the actual operation of the system from this version the H* file is resident on one of the disk packs, along with the data files necessary to perform a retrieval. Thus the retrieval system is completely portable insofar as other HISI-600 systems are concerned. Figure VI.26 is a listing showing the use of the RLHS library program and examples of control cards for the DS-167 packs.

VI.3.6 Illustrations of the Overlay Process

This section contains various charts showing how the LINK (overlay) method of core allocation conserves machine space for the retrieval system. Included are both actual core maps produced by the GELOAD activity which generates the H* file and diagrammatic versions of the program allocations. (Figure VI.27)

VI.4 EXTERNAL FILES USED BY THE DOCUMENT RETRIEVAL SYSTEM

This section discusses each of the various disc-resident files used by the retrieval system, including file organization, diagrams, and estimations of size for other than the present-sized collection of documents.

VI.4.1 STEMDICT

File STEMDICT contains the automatically-generated content-stem dictionary created by the batch system. During the course of a retrieval session STEMDICT is used each time a query is generated or modified; all query words entered by the user are eventually looked up in this file by routine LOOK. File organization for STEMDICT is random, in logical blocks of 60 machine-words each. In turn, each logical block contains ten entries of six machine-words each. Figure VI.27 outlines

```
0001  $       SNUMB   1073T
0002  $       COMMENT BIDBZTAB           TSS CARDIN
0003  $       IDENT   EMIDB,P. KURTZ ,RUNDIA,00199,4594
0004  $       MSG2    1,R.A.D.C/ON-LINE/RETRIEVAL/SYSTEM/IN/USE
0005  $       MSG2    1, PLEASE/DON'T/LET/ME/GET/SWAPPED///THANKS
$$     USERID  BIDBZTAB$####
0007  $       PROGRAM RLHS
0008  $       LIMITS  30,32K,,10000
0009  $$      PRMFL   04,R,S,BIDBZTAB/CONTENTS
0010  $       DAC     05
0011  $       DAC     06
0012  $$      PRMFL   30,R,S,BIDBZTAB/BCDSMS
0013  $$      PRMFL   31,R,S,BIDBZTAB/BCDLMS
0014  $$      PRMFL   DI,R,R,BIDBZTAB/STEMDICT
0015  $       167PK   F1,X1D,R,00006,,,526/30,559/100
0016  $       167PK   F2,X2D,R,00006,,,184/158
0017  $       167PK   F3,X3D,R,00006,,,348/13,364/65
0018  $       167PK   F4,X4D,R,00006,,,441/28,474/49
0019  $       167PK   H*,X5D,R,00006,,,41/56
0020  $       ENDJOB
```

Figure VI. 26. Running the On-Line System from H* Using RLHS

```
  1536T 01  05-14-71    17.344

    ORIGIN   053069       ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION


                                SUBPROGRAMS INCLUDED IN DECK. *** OPTIMIZED LOADER ***

                          $      OPTION  JOVIAL,NOGO,SYMREF,HILOAD                                000001
                          $      USE     .RTYP                                                    000001
    044620   051471       DIALOG 044620
             SYMREFS      FETDOC            FETBIB            MESAGE            SEND              LLINK
                          ISOL              ANAL              LOOK              RETRIV            BINBCD
                          RECV              NUMBER            FETDCV            CMPAIR            .VLTAS
                          .VLTCM            .VENZR            .FEXIT            .VIN              .VIO
                          .VENAS
    040320   042171       MESAGE 040320
             SYMREFS      SEND              .VIN              .VIO              .VLTAS
    040132   031171       BINBCD 040132
             SYMREFS      .VLTAS
    040016   031071       BCDBIN 040016
             SYMREFS      .VLTAS            .VFXEX
    037730   030871       SEND   037730
             SYMREFS      .VOUT             .VIO
    037642   031071       RECV   037642
             SYMREFS      .VIN              .VIO
    037476   030871       HUNT1  037476
    037276   030871       HUNT2  037276
             SYMREFS      .VLTAS
    037232   040771       FETINV 037232
    037166   040771       FETDCV 037166
    037122   040871       FETDOC 037122
    037056   040771       FETBIB 037056


                                SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY.

    037002   110268       .SETU. 037007
             SYMREFS      .FLTPR
    036406   110268       .FOPEN 036412     EXIT   036616     .FEXIT 036616     .FBRD. 036765     .FBFTB 036766
                          .FXOP. 036601     .FGTFB 036406     .FJOV. 036410
             SYMREFS      .FXEM.            .GCLSE            .GOPEN            .GSTIN            .GSTOT
    035124   011970       FXEM   035727     .FXEM. 035124     .FXMC. 035712     ANYERR 036035     FXOPT  035746
                          FXDVCK 036007     FXALT  036023     FXDV   036320     FXFDV  036316     FXCODE 035726
                          .FXSW1 035714     .FXSW2 035716     .FXSW3 035720     ERRLK  036043     .FLTPR 036045
             SYMREFS      .FBAD.            .FEXIT            .FXOP.            .GWTBC
    035006   110268       LINK   035016     LLINK  035014     IDLINK 035037     LENTRY 035007
    034030   081569       .RTYP  034030     .GOPNR 034030     .GCLSR 034112     .GGETR 034262     .GPUTR 034145
                          .GATOB 034600     .GBTOA 034500
             SYMREFS      .GR960
    034000   110268       .GBTIN 034000     SETIN  034000
             SYMREFS      .GR225
    033704   110268       .GSTOT 033704     SETOUT 033704
    033634   073069       .GWTRC 033634     .GAWTP 033634     WTREC  033634
```

Figure VI. 27. Load Map

```
1536T 01  05-14-71   17,344

   ORIGIN    053069      ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

             SYMREFS     .GPUT             .GFR67
   033110    010970      .GCOPY 033110     COPY   033110     .GPTBK 033113     PUTBK  033113     .GPUT  033116
                         PUT    033116     .GACOP 033110     .GAPTB 033113     .GAPUT 033116     .GFR67 033611
             SYMREFS     GET               .GPUTR            .GR200            .GR275            .GR375
                         .GR960
   032372    010271      .GGTBK 032372     GETBK  032372     .GGET  032374     GET    032374     .GAGTB 032372
                         .GAGET 032374
             SYMREFS     .GGETR            .GR225            .GR250            .GR275            .GR375
                         .GR377            .GR960            .GR979            .GR990            .GRCVY
   031630    010970      .GOPEN 031630     .GAOPE 031630     CPEN   031630
             SYMREFS     .GINHD            .GOPNR            .GOUTH            .GR225            .GR960
   031156    073069      .GCLSE 031156     .GACLS 031156     .GR185 031262     .GR186 031353     .GR178 031267
                         CLOSE  031156
             SYMREFS     .GCLSR            .GOUTL            .GR200            .GR275            .GR375
                         .GR960            .GR980
   030764    110268      .GR200 030764
             SYMREFS     .GR275            .GR375            .GR960            .GR980            .GR990
   030706    110268      .GR225 030706
             SYMREFS     .GR979            .GR980            .GR990
   030634    110268      .GR250 030634
             SYMREFS     .GUSWH
   030342    073069      .GR275 030342
             SYMREFS     .GINTL            .GOUTL            .GR390            .GUSWH
   030172    110268      .GR377 030226     .GR385 030165     .GR375 030172     .GR37X 030245     .GR390 030265
             SYMREFS     .GR960            .GR984            .GR985            .GR989
   030066    110268      .GR980 030066     .GR979 030164     .GR99X 030072     .GR984 030130     .GR985 030164
                         .GR999 030076
             SYMREFS     .GR960            .GR991
   030006    110268      .GR960 030006
   027270    110268      .GINRD 027275     .GOUTH 027274     .GINTL 027273     .GOUTL 027272     .GUSWH 027271
                         .GOVRL 027276     .GLREA 027354     .GBCVY 027270
             SYMREFS     .GINID
   027236    110268      .GR990 027236     .GR991 027257     15AUG5 027264
   027234    110268      .GINID 027234
   026432    010970      .VIN   026432     .VOUT  026453     .VIO   026467
             SYMREFS     .GGET             .GWEF             PUNCH             .FBAD.            .FBST.
                         .FJOV.            .FXEM.            .GBSRC            .GCLBE            .GFRCE
                         .GOPEN            .GSTOT            .GWTRC
   026114    010271      .FBST. 026114
             SYMREFS     .GGET             .FBAD.            .FGTFB            .FXEM.            .GBSRC
   025664    110268      .GBSRC 025664     .GABSR 025664     BSRBC  025664
             SYMREFS     .GR375            .GR960            .GRLSE
   025456    110268      .GWEF  025456     .GAWEF 025456     WEF    025456
             SYMREFS     .GR200            .GR275            .GR960
   025344    110268      .GPNCH 025344     .GAPNC 025344     PUNCH  025344
             SYMREFS     .GE068            .GILLB            .GWTRC
   025024    110268      .GIL01 025204     .GILLB 025024
             SYMREFS     .GBCD             .GBMRY            .GE065            .GE066
   024710    110268      .GEDIT 024710     .GE062 025003     .GE063 025004     .GE064 025005     .GE065 025007
                         .GE066 025010     .GE067 025011     .GAEDI 024710     .GE068 025012     .GE069 025013
```

Figure VI.27. (Continued)

```
 1536T 01  05-14-71   17,344

   ORIGIN   053069      ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION

                        .GF071 025014    .GF072 025015    EDATE  025016    ETIME  025017    IOEDIT 024710
            SYMREFS     .GRCD
   024624   073069      .GFRCE 024624    .GAFRC 024624    FORCE  024624
            SYMREFS     .GOUTL           .GR960           .GRLSE           .GUSWH
   024524   073069      .GRLSE 024524    .GARLS 024524    RELSE  024524
            SYMREFS     .GR200           .GR275           .GR375           .GR960
   024506   110268      .GBNRY 024506
   024460   110268      .GBCD  024460
   024332   110268      .VENAS 024332
   024246   110268      .VFXEX 024246
            SYMREFS     .VEXP
   024200   110268      .VENZR 024200
   023626   110268      .VLTAS 023626
            SYMREFS     .VBTPT           .VBTPT
   023126   110268      .VLTCM 023126
            SYMREFS     .VBTPT
   022754   110268      .VBTPT 022754
   022564   073069      .VBTPT 022564
   022516   110268      .VEXP  022516
            SYMREFS     .FDXP2
   022376   010271      .FDXP2 022376
            SYMREFS     DEXP             DLOG             .MXNO            .FXEM.           .MMXNO
   022154   010271      DLOG10 022154    DLOG   022160
            SYMREFS     .MXNO            .FXEM.           .MMXNO
   022026   010271      DEXP   022031
            SYMREFS     .MXNO            .FXEM.           .MMXNO
   022022   010271      .MXNO  022022    .MMXNO 022023
                                              RANGE            SIZE
                        ALLOCATED CORE   000000 THRU 073777     074000
                        OBJECT PROGRAM
                           RELOCATABLE   022020 THRU 073777     051760

                        $      LINK    ISOLAT                                               000003

                               SUBPROGRAMS INCLUDED IN DECK, *** OPTIMIZED LOADER ***

   021440   032271      ISOL   021440
            SYMREFS     RECV             SEND             .VLTAS
                        $      ENTRY   ISOL                                                 000003

                               SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY,

                                              RANGE            SIZE
                        ALLOCATED CORE   000000 THRU 073777     074000
                        OBJECT PROGRAM
                           RELOCATABLE   021440 THRU 073777     052340
 *** NON FATAL ERROR * MISSING ROUTINE   ANAL
 *** NON FATAL ERROR * MISSING ROUTINE   LOOK
```

Figure VI. 27. (Continued)

```
 15361 01  05-14-71   17.344
   ORIGIN   053069       ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

  *** NON FATAL ERROR * MISSING ROUTINE   RETRIV
  *** NON FATAL ERROR * MISSING ROUTINE   NUMBER
  *** NON FATAL ERROR * MISSING ROUTINE   CWPAIR

                         $      LINK    ANALYZ,ISOLAT                                                      000003

                                         SUBPROGRAMS INCLUDED IN DECK. *** OPTIMIZED LOADER ***

   020010    051471      ANAL   020010
             SYMREFS     HUNT1             HUNT2              .VLTAS             .VLTCM                    .VLTEX
                         $      ENTRY   ANAL                                                               000003

                                         SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY.

   017400    110268      .VLTEX 017400
             SYMREFS     .VBTPT           .VBTPT
                                             RANGE               SIZE
                         ALLOCATED CORE    000000 THRU 073777    074000
                         OBJECT PROGRAM
                            RELOCATABLE    017400 THRU 073777    054400
  *** NON FATAL ERROR * MISSING ROUTINE   LOOK
  *** NON FATAL ERROR * MISSING ROUTINE   RETRIV
  *** NON FATAL ERROR * MISSING ROUTINE   NUMBER
  *** NON FATAL ERROR * MISSING ROUTINE   CWPAIR

                         $      LINK    CWPARE,ANALYZ                                                      000003

                                         SUBPROGRAMS INCLUDED IN DECK. *** OPTIMIZED LOADER ***

   021132    031171      CWPAIR 021132
             SYMREFS     RECV              BCDBIN             SEND               .VLTAS                    000004
                         $      ENTRY   CWPAIR

                                         SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY.

                                             RANGE               SIZE
                         ALLOCATED CORE    000000 THRU 073777    074000
                         OBJECT PROGRAM
                            RELOCATABLE    021132 THRU 073777    052646
  *** NON FATAL ERROR * MISSING ROUTINE   LOOK
  *** NON FATAL ERROR * MISSING ROUTINE   RETRIV
  *** NON FATAL ERROR * MISSING ROUTINE   NUMBER

                         $      LINK    NUMBAN,CWPARE                                                      000004
```

Figure VI.27. (Continued)

```
1536T 01  05-14-71   17.344

  ORIGIN   053059      ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                                   SUBPROGRAMS INCLUDED IN DECK. *** OPTIMIZED LOADER ***

  020506   050471      NUMBER 020506
           SYMREFS     RECV              BCDBIN            SEND              .VLTAS            .VLTCM
                       $      ENTRY   NUMBER                                                   000004

                                   SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY.

                                              RANGE              SIZE
                       ALLOCATED CORE    000000 THRU 073777      074000
                       OBJECT PROGRAM
                          RELOCATABLE    020506 THRU 073777      053272
*** NON FATAL ERROR * MISSING ROUTINE    LOOK
*** NON FATAL ERROR * MISSING ROUTINE    RETRIV

                       $      LINK    LOOKUP,NUMBAH                                            000004

                                   SUBPROGRAMS INCLUDED IN DECK. *** OPTIMIZED LOADER ***

  020322   051471      LOOK   020322
           SYMREFS     OPENDI            HUNT2A            RDBUCK            HUNT2B            CLOSDI
                       MESAGE            SEND              .VLTAS            .VLTCM
                       $      ENTRY   LOOK                                                     000004
  020224   040171      RDBUCK 020263     OPENDI 020225     CLOSDI 020251
           SYMREFS     OPEN              READ              WAIT              CLOSE
  020006   032471      HUNT2A 020006
           SYMREFS     .VLTAS
  017606   032471      HUNT2B 017606
           SYMREFS     .VLTAS

                                   SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY.

  017512   110268      .GREAD 017512     .GARBA 017512     READ   017512
           SYMREFS     .GR250            .GR960            .GR980            .GR980            .GR99X
  017430   110268      .GWAIT 017430     .GAWAI 017430     WAIT   017430
           SYMREFS     .GR275            .GR375            .GR37X            .GR960
                                              RANGE              SIZE
                       ALLOCATED CORE    000000 THRU 073777      074000
                       OBJECT PROGRAM
                          RELOCATABLE    017430 THRU 073777      054350
*** NON FATAL ERROR * MISSING ROUTINE    RETRIV

                       $      LINK    RETRIN,LOOKUP                                            000005

                                   SUBPROGRAMS INCLUDED IN DECK. *** OPTIMIZED LOADER ***
```

Figure VI. 27. (Continued)

```
(   1536T 01  05-14-71    17.344

    ORIGIN    053069       ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION
( -
    007560    050571       RETRIV 007560
              SYMREFS      SEND              RETINV            RETDCV            .VXPNT
(                          $      ENTRY   RETRIV                                                   000005

(                                SUBPROGRAMS OBTAINED FROM SYSTEM  LIBRARY.

    007534    110266       .VXPNT 007534
(             SYMREFS      .VEXP
                                                RANGE               SIZE
                           ALLOCATED CORE    000000 THRU 073777     074000
(                          OBJECT PROGRAM
                              RELOCATABLE    007534 THRU 073777     064244
                           $        PRMFL   04,R,S,BIDBZTAB/CONTENTS                               000005
(                          $        DAC     05                                                     000005
                           $        DAC     06                                                     000005
                           $        PRMFL   30,R,S,BIDBZTAB/BCDSMS                                 000005
(                          $        PRMFL   31,R,S,BIDBZTAB/BCDLMS                                 000006
                           $        PRMFL   DI,R,R,BIDBZTAB/STEMDICT                               000006
                           $        167PK   F1,X1D,R,00006,,,526/30,559/100                        000006
(                          $        167PK   F2,X2D,R,00006,,,184/158                               000006
                           $        167PK   F3,X3D,R,00006,,,348/13,364/65                         000006
                           $        167PK   F4,X4D,R,00006,,,441/28,474/49                         000006
(                          $        167PK   H*,X5D,R,00006,,,41/56                                 000006

                           FCB AND BUFFER SPACE
(
                              AVAILABLE        000101 THRU 007531     007431
                              FILE CTRL BLKS 007224 THRU 007532       000307
(                             MAXIMUM BUFFER SPACE REQUIRED           004300

                               29K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN
(
                           EXECUTION   PROGRAM ENTERED AT      044620

(                          THERE WERE  000015 WARNING FLAGS IN THE ABOVE LOAD

(

(

(

(
```

Figure VI. 27. (Concluded)

the logical structure of the file as a whole. Details of the LOOKup process are to be found in Section VI. 2. 3. 4 of this report.

STEMDICT occupies five hundred logical blocks of 64 machine-words each. Although the data structure only occupies 60 out of the 64 available machine-words per block, this addressing method vastly simplifies access to the file, thus reducing software overhead with a very small sacrifice of disc space. The amount of space cited above has been allocated for a dictionary of 5000 content stems. Each increase of ten stems in the dictionary requires one additional 64-word block. STEMDICT is small enough to reside either on a dedicated disk pack or within the GECOS-III file system permfiles.

VI. 4. 2 CONTENTS

The two-level dictionary search algorithm requires a table of contents consisting of the last stem-dictionary entry in each of the 500 disk buckets. File CONTENTS contains this information. The organization of CONTENTS is sequential, standard GECOS-III system format; file size is 500 entries, two machine-words per entry, plus any additional block and record control information placed there by GEFRC (General File and Record Control routines). One additional entry must be made in CONTENTS for each ten new stems added to file STEMDICT. CONTENTS is small enough to reside totally within the permanent file system under the user's master catalog name. CONTENTS is loaded at on-line system initialization by procedure LODCON.

VI. 4. 3 BCDSMS : BCDLMS

These two files are the small and large user message files, respectively. At on-line initialization time BCDLMS (the verbose messages) are by default loaded in the MESAGE table area. In response to a request by the user for the opposite message form, the appropriate type of file is accessed and the messages loaded. BCDSMS and BCDLMS are stored as sequential access standard system format files. The size of these files varies, depending upon the actual message texts stored there. Currently, a maximum of 6 standard blocks is allocated for each file; the internal space allocated within MESAGE is a table of 150 entries of 12 machine-words per entry.

VI. 5 DETAILED DESCRIPTION OF THE DIALOGUE PROCESSOR EXECUTIVE

In the sub-sections are described those portions of the on-line system not yet described in detail. Included here are all those core-resident portions of the system compiled with the main program, as well as the main program itself. Detailed descriptions are given for all tables, arrays, variables (when important), and flags. Also included is a master flowchart and flowcharts for the processor main program.

VI. 5. 1 Table Descriptions

Table Name	Item Names	Table Function	Item Functions
QCVF		compact query-concept-vector	
	QCVACN		concept number
	QCVCOR		correlation (weight) of this concept
QUERST		query-stem table	
	QURENT		query stem entry
	QUPAIRS		pairs of concepts and weights for this query stem
CONTENTS		table-of-contents for disc buckets of stem dictionary	
	CONT		end-of-bucket stem value
NONOT		document exclusion list	
	NONO		accession number of excluded document

Table Name	Item Names	Table Function	Item Functions
NOTAB		rejected words table	
	NNOT		copy of rejected word
TMPLN		temp table print-out	
	NAME		header names
	DATSTF		temp data in BCD
TEMP		retrieval temporary table	
	ACCNUM		accession number of retrieved document
	TEMPID		temporary ID number of retrieved document
	RETRNK		rank of document when last retrieved
	BIBSUP		suppress bibliographic data printing flag
	LASRET		document retrieved in last session flag
	CORCOE		correlation between this document and the QCV when last retrieved
QCV		query-concept-vector	
	CORREL		correlation measure (QCV index is the implicit concept number)
WORDS		query-words table	
	WRDENT		copy of the query word entry
STEMS		query-stems table	
	STMENT		copy of the query stem entry

Table Name	Item Names	Table Function	Item Functions
DELWDS		words-to-delete table	
	DELWD		copy of word to be deleted entry
DELCNTS		character-count of words-to-be-delete	
	DELCNTR		character-count value
OPTNAM		BCD names of the user functions	
	NAMES		BCD names of mnemonics for user functions
COUNTS		character-count of query-words	
	COUNTER		character-count value
MODES		flag-value-table for system modes	
	MODEF		mode setting value

Array Name	Array Function
DCV	Document-concept-vector copy

VI. 5. 2 Global Variables Description

Variable Name	Variable Function
TERSE	(overlaid into the MODES table) terse-verbose dialogue switch
SKIP1	(overlaid into the MODES table) skip initial query formation flag
PRINTQ	(overlaid into the MODES table) print query data flag
PRINTR	(overlaid into the MODES table) print retrieval data flag
OPTION	(overlaid into the MODES table) assume sophisticated user flag
PRIVIT	(overlaid into the MODES table) allow user "privity" into the on-line system; flag dummy in operational version of the system
MODKNT	(overlaid into the MODES table) dummy flag
NEWQ	allow new query to be started flag
WFLG	word-modification in use flag
RFLG	retrieve-active flag
SEEFLG	SEE option used flag
DEFLG	DEL option used flag (DELete used during MOD)
WRDFLG	WRD option used flag (WoRD used during MOD)
DOCDOC	DDC option used flag (DocDoc retrieval used)
NULQRY	null query flag
OFFLG	OFFline data generated (unimplemented)
PREFIL	PREserve offline data flag (unimplemented)
ACORTM	Accession or temporary ID flag

Variable Name	Variable Function
INVALFLG	concept-weight pair was invalid flag
PS	printing of the current bibliography was suppressed

VI. 6 FUTURE DIRECTION OF THE ON-LINE SYSTEM

In light of the modular design of the on-line system it is feasible to consider developing it further, along lines to be summarized in the paragraphs below. The several areas to be discussed include the following topics: additional program segmentation; adding new functions to the system; increasing the user's flexibility and control over the operation of the system.

VI. 6. 1 Further Program Segmentation

The guidelines for segmentation of the Dialogue Processor have already been laid out (as discussed in Section VI. 2. 3 of this report). Further segmentation is not only possible but quite desirable from the standpoint of core space conservation. The logic of the processor is such that only one user function is in use at any one time, although control is frequently passed from one function to another. No one of the functions is particularly large by itself but, as in the case of the already overlaid portions of the system, the sum total of all the functions adds up to a reasonably large amount of core space.

The logical extension to this discussion is segmentation of the processor by individual user functions, as well as those portions which have previously been included in the segmentation scheme. However, there is an additional problem when considering the functions that is not present in the case of the presently overlaid portions of the system. Individiual functions quite frequently pass control to others, especially when a "tyro" user is in control of the system. In this case the system logic calls all the necessary functions for the user, except for END, DOC, and MOD.

Because of the above arrangement, many of the user functions have multiple entry points in their code. It, therefore, is not enough to separate all the functions into separately-compiled routines and load and execute them when needed. A method must also be supplied whereby each function with more than one entry point may be entered at various points and in turn may exit to different points. The key to the solution is to expand the OPT function which, in the present implementation, handles explicit calls to the functions. The new OPT function would handle calls to user functions by the BCD-encoded name of the function, rather than directly through the switch table as in the present implementation. Also, a flag would have to be passed indicating whether or not a secondary entry-point was desired and if so which one. If the above plan were followed, all of the user functions could be implemented as separately-compiled routines in the fashion of the present overlaid portions of the system.

VI. 6. 2 How to Create Additional Functions

As presently implemented, each user function consists of three portions: an entry into the "option-names" table of a BCD mnemonic representative of the function; a corresponding branch in the controlling switch table consisting of the actual JOVIAL label-name for the function entry; the actual code for performing the function, with an entry-label as specified in the control switch table.

In order to add new functions to the present system the following steps are necessary: design, code, and test the function program and include in the body of the Dialogue Processor the already tested code necessary to perform the desired function, increase the count of available options (the table declaration for OPTNAM). If the revised version of OPT is also implemented (for user function segmentation), the increase in options may have to be taken into account. In addition, in the case of revised OPT, new $ LINK control cards will have to be added to the permfile from which the final version of the executable system is generated onto dispack.

VI.6.3 Increasing the User's Control Over the System

One desirable feature of an on-line system which has not been implemented in the present version is the ability to interrupt the processing at any time and return to a different level in the code. Telecommunications terminals usually have "break" or "interrupt" keys which may be used for this purpose. However, in order to take advantage of this feature utilizing GECOS-III, the telecommunications software interface within the Dialogue Processor must be changed.

The present implementation uses conversational GEFRC calls, as these are the simplest to program and the easiest to use from the user standpoint. Ultimately, however, GEFRC must issue its own calls to GERTS in order to connect to the remote terminal. Through the use of MME GEROUT calls, any batch-world program may communicate with a remote device. Of course, some of the advantages of conversational GEFRC are lost when direct calls are made: the program must handle BCD to ASCII and ASCII to BCD conversions; terminal status must be constantly checked for error conditions; connection and disconnection must be handled explicitly. On the other hand, several advantages accrue through the use of direct GEROUT calls: a certain number of library routines are not loaded, lightening the core load; special control characters may be sent to the terminal, a feature not possible through conversational GEFRC; "break" or "interrupt" processing may be implemented. The latter feature is especially important in the case of an on-line system.

With the rather large amount of bibliographic and abstract data being sent to the user, the occasion very frequently may arise when it is desirable to interrupt sending of the rest of an abstract, either because the user decides that he has seen enough or because he wants to pursue some other option. In these cases implementation of "break" processing is the natural method for expanding user control over the system.

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROGRM DIALOG                                                                              00000080        0001
  0000      START                                                                          00000085        0002
  0000      '' ******** DIALOGUE PROCESSOR -- VERSION MAY 13,1971 ******** ''              00000140        0003
  0000                                                                                     00000150        0004
  0000      ''MODES''                                                                      00000160        0005
  0001      TABLE MODES R 6 N$                                                             00000170        0006
  0001       BEGIN                                                                         00000180        0007
  0002        ITEM MODEF S V(OFF) V(ON) $                                                  00000190        0008
  0003       END                                                                           00000200        0009
  0003      ITEM TERSE  S V(OFF) V(ON) $                                                   00000210        0010
  0004      ITEM SKIP1  S V(OFF) V(ON) $                                                   00000220        0011
  0005      ITEM PRINTQ S V(OFF) V(ON) $                                                   00000230        0012
  0006      ITEM PRINTR S V(OFF) V(ON) $                                                   00000240        0013
  0007      ITEM OPTION S V(OFF) V(ON) $                                                   00000250        0014
  0008      ITEM PRIVIT S V(OFF) V(ON) $                                                   00000260        0015
  0009      ITEM MODKNT S V(OFF) V(ON) $                                                   00000270        0016
  0009      ''OTHER STATUS FLAGS''                                                         00000280        0017
  0010      ITEM NEWQ   S V(OFF) V(ON) $                                                   00000290        0018
  0011      ITEM WFLG   S V(OFF) V(ON) $                                                   00000300        0019
  0012      ITEM RFLG   S V(OFF) V(ON) $                                                   00000310        0020
  0013      ITEM SEEFLG S V(OFF) V(ON) $                                                   00000320        0021
  0014      ITEM DEFLG  S V(OFF) V(ON) $                                                   00000330        0022
  0015      ITEM WRDFLG S V(OFF) V(ON) $                                                   00000340        0023
  0016      ITEM DOCDOC S V(OFF) V(ON) $                                                   00000350        0024
  0017      ITEM NULQRY S V(OFF) V(ON) $                                                   00000360        0025
  0018      ITEM OFFLG S V(UNUSED) V(USED)$                                                00000370        0026
  0019      ITEM PREFIL S V(NO) V(YES)$                                                    00000380        0027
  0020      ITEM ACOBTM S V(AC) V(TM)$                                                     00000390        0028
  0020      ''MISC, ITEM DECLARATIONS''                                                    00000391        0029
  0021      ITEM ICOR I 36 S $                                                             00000392        0030
  0022      ITEM HICOR H 6 $                                                               00000393        0031
  0023      OVERLAY HICOR=ICOR $                                                           00000394        0032
  0024      ITEM CHATH H 6 $                                                               00000400        0033
  0025      ITEM HITS     I 36 S $                                                         00000410        0034
  0026      ITEM CHGBPOS I 36 S $                                                          00000420        0035
  0027      ITEM CHGWRD  I 36 S $                                                          00000430        0036
  0028      ITEM CHGCHR  I 36 S $                                                          00000440        0037
  0029      ITEM CHBYTE H 6 $                                                              00000441        0038
  0030      OVERLAY CHBYTE=CHGCHR$                                                         00000450        0039
  0031      ITEM DOCUNM  I 36 S $                                                          00000451        0040
  0032      ITEM DOCUNOH H 6 $                                                             00000460        0041
  0033      ITEM DDCCON  I 36 S $                                                          00000470        0042
  0034      ITEM DDCWGT A 18 S 14$                                                         00000480        0043
  0035      ITEM PLACE H 6$                                                                00000490        0044
  0036      ITEM MODESTR H 12$                                                             00000500        0045
  0037      ITEM DDCWLD  I 36 S $                                                          00000510        0046
  0038      ITEM WGTCON   H 6 $                                                            00000520        0047
  0039      ITEM WGTCOR   I 36 S $                                                         00000521        0048
  0040      ITEM WGTCORH H 6 $                                                             00000522        0049
  0041      OVERLAY WGTCOR=WGTCORH$
```

Figure VI. 28. DIALOGUE Processor Listing

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0042      ITEM CWFIELD H 12 P 12H(             )$                     00000530     0050
  0043      ITEM WGTCORF  F $                                           00000540     0051
  0044      ITEM WGTCONCEP I 36 S $                                     00000550     0052
  0045      ITEM WGTCONCEPH H 6 $                                       00000551     0053
  0046      ITEM WGTWEIGHT F $                                          00000560     0054
  0047      ITEM INVALFLG S V(NO) V(YES) $                              00000570     0055
  0048      ITEM NUMB      H 6 $                                        00000580     0056
  0049      ITEM DOCCORF F $                                            00000590     0057
  0050      ITEM DOCCOR   I 36 S $                                      00000600     0058
  0051      ITEM DOCCORH H 6 $                                          00000601     0059
  0052      ITEM PS         S V(OFF) V(ON) $                            00000610     0060
  0053      ITEM RESET I 36 S $ ''RESET "NUMBER" AND RETURN INDICATOR'' 00000620     0061
  0054      ITEM CONPAIRS H 18 $                                        00000630     0062
  0055      ITEM CONCONCEPT I 36 S $                                    00000640     0063
  0056      ITEM CONWEIGHT I 36 S $                                     00000650     0064
  0057      ITEM DONE B $                                               00000651     0065
  0058      ITEM CONCONCEPTH H 6 $                                      00000652     0066
  0059      ITEM CONWEIGHTH H 6 $                                       00000653     0067
  0060      ITEM TEMPH H 6 $                                            00000654     0068
  0061      ITEM NEWTID I 36 S P 1 $                                    00000655     0069
  0062      ITEM DCVHLD I 36 S $                                        00000657     0070
  0062      ''                                                          00000660     0071
  0062        TABLES,ARRAYS AND MESSAGES (EVENTUALLY)''                 00000670     0072
  0063      TABLE QCVF V 200 S 2 $                                      00000680     0073
  0063       BEGIN                                                      00000681     0074
  0064        ITEM QCVACH I 36 S 0 0 $                                  00000682     0075
  0065        ITEM QCVCOR F       1 0 $                                 00000683     0076
  0066       END                                                        00000684     0077
  0066      TABLE QUEEST V 100 S 6 $                                    00000690     0078
  0066       BEGIN                                                      00000700     0079
  0067        ITEM QUEENT H 12 0 0$                                     00000710     0080
  0068        ITEM QUPAIRS H 24 2 0 $                                   00000720     0081
  0069       END                                                        00000730     0082
  0069      TABLE CONTENTS V 500 S 2 $                                  00000740     0083
  0071       BEGIN ITEM CONT H 12 0 0 $ END                             00000741     0084
  0071      TABLE NONOT V 100 S 1$                                      00000750     0085
  0071       BEGIN                                                      00000760     0086
  0072        ITEM NONO I 18 U 0 18 $                                   00000770     0087
  0073       END                                                        00000780     0088
  0073                                                                  00000790     0089
  0073      TABLE NOTAB V 100 S 3$                                      00000800     0090
  0073       BEGIN                                                      00000810     0091
  0074        ITEM NNOT H 18 0 0$                                       00000820     0092
  0075       END                                                        00000830     0093
  0075                                                                  00000840     0094
  0075      TABLE TMPLN R 6 S 3 $                                       00000850     0095
  0075       BEGIN                                                      00000860     0096
  0076        ITEM NAME H 6 0 0 $                                       00000870     0097
  0076        BEGIN 6H(ACCES#)                                          00000880     0098
  0076              6H(TEMP# )                                          00000890     0099
```

Figure VI.28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0076               6H(RANK  )                                                 00000900          0100
 0076               6H(SUPPRS)                                                 00000910          0101
 0076               6H(LASRET)                                                 00000920          0102
 0076               6H(CORREL)                                                 00000930          0103
 0076          END                                                             00000940          0104
 0077          ITEM DATSIF H 12 1 0 $                                          00000950          0105
 0077          BEGIN 12H(            )                                         00000960          0106
 0077                12H(            )                                         00000970          0107
 0077                12H(            )                                         00000980          0108
 0077                12H(            )                                         00000990          0109
 0077                12H(            )                                         00001000          0110
 0077                12H(  .         )                                         00001010          0111
 0077          END                                                             00001020          0112
 0078          ITEM TMPFAKE H 18 0 0 $                                         00001025          0113
 0079        END                                                               00001030          0114
 0079                                                                          00001040          0115
 0079        TABLE TEMP V 256 S 2 $                                            00001050          0116
 0079         BEGIN                                                            00001060          0117
 0080          ITEM ACCNUM I 16 U 0  0$                                        00001070          0118
 0081          ITEM TEMPID I  9 U 0 16$                                        00001080          0119
 0082          ITEM RETRNK I  7 U 0 25$                                        00001090          0120
 0083          ITEM BIBSUP I  1 U 0 32$                                        00001100          0121
 0084          ITEM LASRET I  1 U 0 33$                                        00001110          0122
 0085          ITEM CORCOE F       1  0$                                       00001120          0123
 0086         END                                                              00001130          0124
 0086        TABLE QCV R 1502 S 1 $                                            00001150          0125
 0086         BEGIN                                                            00001160          0126
 0087          ITEM CORREL F 0 0 $                                             00001170          0127
 0088         END                                                              00001180          0128
 0088                                                                          00001190          0129
 0088        TABLE WORDS V 100 S 3$                                            00001200          0130
 0088         BEGIN                                                            00001210          0131
 0089          ITEM WRDENT H 18 0 0$                                           00001220          0132
 0090         END                                                              00001230          0133
 0090        ITEM LINEND I 36 S P 0(000000007701)$                             00001240          0134
 0091        TABLE STEMS V 100 S 2$                                            00001250          0135
 0091         BEGIN                                                            00001260          0136
 0092          ITEM STMENT H 12 0 0$                                           00001270          0137
 0093         END                                                              00001280          0138
 0093                                                                          00001290          0139
 0093        TABLE DELWDS V 100 S 3$                                           00001300          0140
 0093         BEGIN                                                            00001310          0141
 0094          ITEM DELWD H 18 0 0$                                            00001320          0142
 0095         END                                                              00001330          0143
 0095                                                                          00001340          0144
 0095        TABLE DELCNTS V 100 S 1 $                                         00001341          0145
 0095         BEGIN                                                            00001342          0146
 0096          ITEM DELCNTR I 36 S 0 0 $                                       00001343          0147
 0097         END                                                              00001344          0148
 0097        TABLE OPTNAM R 17 S 1$                                            00001350          0149
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0097       BEGIN ''USER OPTIONS LIST DEFINITION''                        00001360            0150
  0098        ITEM NAMES H 6 0 0 $                                         00001370            0151
  0098         BEGIN                                                       00001380            0152
  0098          6H(CHG   )                                                 00001390            0153
  0098          6H(DDC   )                                                 00001400            0154
  0098          6H(DQC   )                                                 00001410            0155
  0098          6H(RET   )                                                 00001420            0156
  0098          6H(WRD   )                                                 00001430            0157
  0098          6H(CLR   )                                                 00001440            0158
  0098          6H(DEL   )                                                 00001450            0159
  0098          6H(MOD   )                                                 00001460            0160
  0098          6H(SEE   )                                                 00001470            0161
  0098          6H(CON   )                                                 00001480            0162
  0098          6H(ENDF  )                                                 00001490            0163
  0098          6H(OFF   )                                                 00001491            0164
  0098          6H(WGT   )                                                 00001510            0165
  0098          6H(ENDRUN)                                                 00001520            0166
  0098          6H(CHAT  )                                                 00001530            0167
  0098          6H(BLD   )                                                 00001531            0168
  0098          6H(SKIM  )                                                 00001532            0169
  0098         END                                                         00001540            0170
  0099        END                                                          00001550            0171
  0099       ITEM WRDSTEM H 6 P                                            00001560            0172
  0099           36H(                                 )$                   00001561            0173
  0100       ITEM TEMPLINE H 66 $                                          00001570            0174
  0101       ITEM SKINESS H 24 P 24H(       ENTRIES IN "TEMP")$            00001571            0175
  0102       TABLE COUNTS V 100 S 1 $                                      00001610            0176
  0102        BEGIN                                                        00001611            0177
  0103         ITEM COUNTER I 36 S 0 0 $                                   00001612            0178
  0104        END                                                          00001613            0179
  0104       TABLE DCV V 256 S 1 $                                         00001620            0180
  0104        BEGIN                                                        00001630            0181
  0105         ITEM DCVCON I 18 S 0 0 $                                    00001640            0182
  0106         ITEM DCVWGT A 18 S 14 0 18 $                                00001650            0183
  0107         ITEM DCVPCT I 18 S 0 18 $                                   00001660            0184
  0108        END                                                          00001670            0185
  0109       ITEM DDDDDD I 36 S $ ITEM DDDD I 36 S $                       00001680            0186
  0110       OVERLAY DCV=DDDD,DDDDDD$                                      00001690            0187
  0111       FILE CONTEN V(NORM) V(NULL) V(ENG) V(EOF) NOU$                00001700            0188
  0112       OVERLAY MODES=                                                00002000            0189
  0112       MODKNT,TERSE,SKIP1,                                           00002010            0190
  0112       PRINTQ,PRINTR,OPTION,PRIVIT$                                  00002020            0191
  0113       SWITCH OPTSW=                                                 00002030            0192
  0113       (CHG,DDC ,DOC,                                                00002040            0193
  0113        RET,WRD ,CLR,                                                00002050            0194
  0113        DEL,MOD ,SEE,                                                00002060            0195
  0113        CON,ENDF,OFF,                                                00002070            0196
  0113        WGT,JOBEND,CHAT,BLD,SKIM)$                                   00002080            0197
  0114       DIALOG. MESAGE(-1)$''LOAD UP VERBOSE MESSAGES TO BEGIN''      00002100            0198
  0115       FOR I=ALL(MODES)$                                             00002119            0199
```

Figure VI.28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0116        MODEF($I$)=V(OFF)$''CLEAR ALL OPTIONS''                                   00002120            0200
  0117       MESAGE(74)$                                                                00002125            0201
  0118       LODCONS''LOAD UP TABLE-OF-CONTENTS FOR STEM DICTIONARY''                   00002126            0202
  0119       SEND(18H(*VERSION--05/13/71),3,18)$                                        00002127            0203
  0119       ''                                                                         00002130            0204
  0119       NOW CLEAR ALL OTHER FLAGS TO 'OFF'                                         00002140            0205
  0119       ''                                                                         00002150            0206
  0120       NEWQ  =V(OFF)$                                                             00002160            0207
  0121       WFLG  =V(OFF)$                                                             00002170            0208
  0122       RFLG  =V(OFF)$                                                             00002180            0209
  0123       SRRFLG=V(OFF)$                                                             00002190            0210
  0124       DEFLG =V(OFF)$                                                             00002200            0211
  0125       WRDFLG=V(OFF)$                                                             00002210            0212
  0126       DOCDOC=V(OFF)$                                                             00002220            0213
  0127       NULQRY=V(OFF)$                                                             00002230            0214
  0128       OFFLG=V(UNUSED)$                                                           00002240            0215
  0129       PREFIL=V(NO)$                                                              00002250            0216
  0129       ''                                                                         00002260            0217
  0129       CLEAR OUT ALL TABLES TO THEIR INITIAL STATES                               00002270            0218
  0129       ''                                                                         00002280            0219
  0131       FOR I=0,1,255$ BEGIN                                                       00002290            0220
  0132         ACCNUM($I$)=0$                                                           00002300            0221
  0133         TEMPID($I$)=0$                                                           00002310            0222
  0134         RETRNK($I$)=0$                                                           00002320            0223
  0135         BIBSUP($I$)=0$                                                           00002330            0224
  0136         LASRET($I$)=0$                                                           00002340            0225
  0137       CORCOR($I$)=0,0$                                                           00002350            0226
  0138        END                                                                       00002360            0227
  0139       FOR I=0,1,1501$                                                            00002370            0228
  0140        BEGIN                                                                     00002380            0229
  0141         CORREL($I$)=0,0$                                                         00002400            0230
  0142        END                                                                       00002410            0231
  0143       FOR I=0,1,99$                                                              00002420            0232
  0144        BEGIN                                                                     00002430            0233
  0145         WRDENT($I$)=18H(                  )$                                     00002440            0234
  0146         STMENT($I$)=12H(            )$                                           00002450            0235
  0147         QUEENT($I$)=12H(            )$                                           00002460            0236
  0148         NONO($I$)=0$                                                             00002470            0237
  0149         NNOT($I$)  =18H(                  )$                                     00002480            0238
  0150        END                                                                       00002490            0239
  0150       ''                                                                         00002500            0240
  0150       NOW START INTO THE GUTS OF THINGS IN THE PROCESSOR                         00002510            0241
  0150       ''                                                                         00002520            0242
  0151       DIA0, MESAGE(7)$                                                           00002530            0243
  0152       SWITCH DIAA=(DIA1,DIA6,OPT)$                                               00002540            0244
  0153       SORCH=YESNO(XXX)$                                                          00002541            0245
  0154       GOTO DIAA($SORCH$) $                                                       00002542            0246
  0154       ''                                                                         00002550            0247
  0154       CHECK FOR MODE 2''                                                         00002560            0248
  0155       DIA1,                                                                      00002570            0249
```

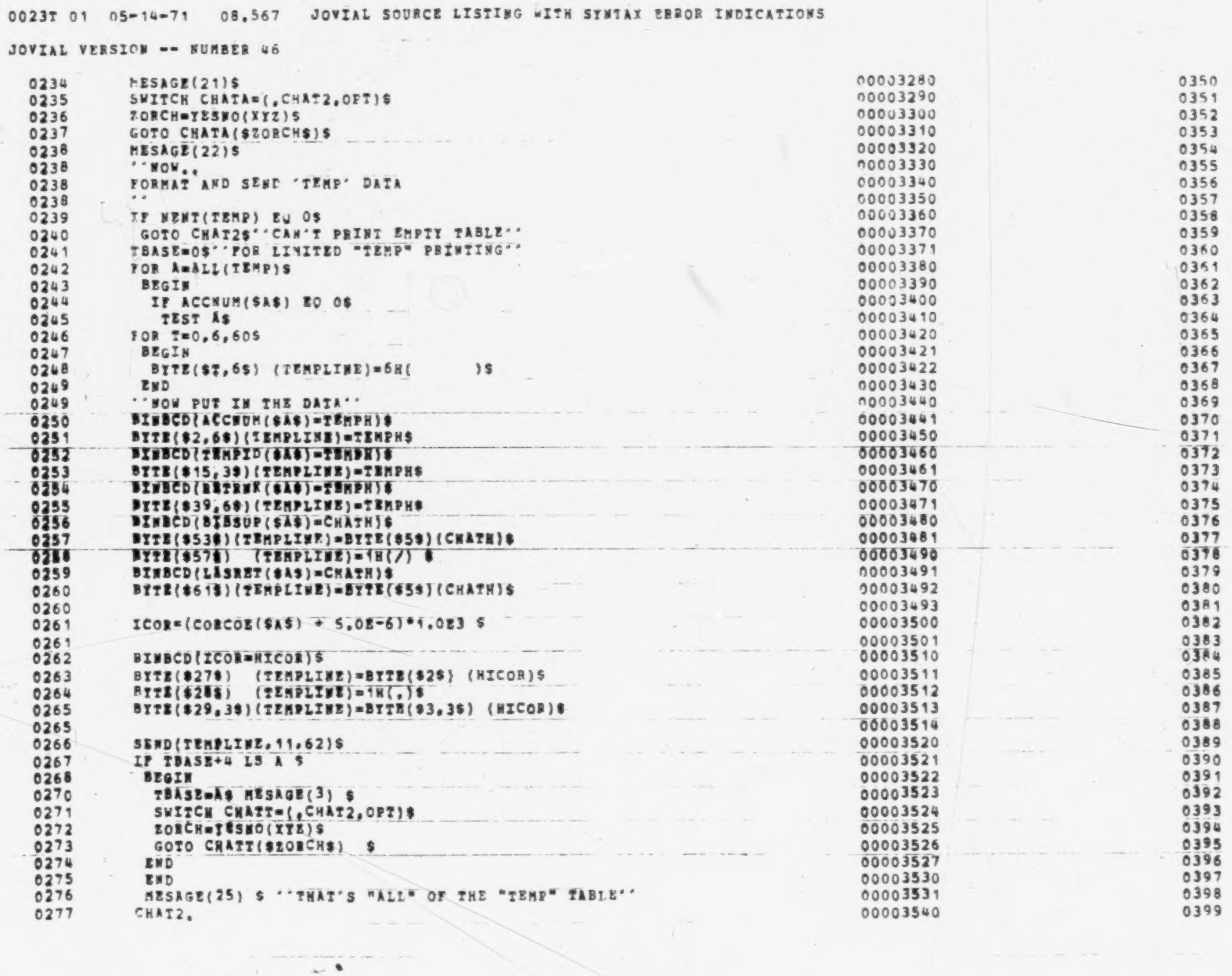
Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- FUMBER 46

 0155      IF SKIP1 EQ V(CN)$                                                00002580          0250
 0156        GOTO DIAR$                                                      00002590          0251
 0157      MESAGE(9)$                                                        00002600          0252
 0158      DIA2, NENT(WORDS)=0$                                              00002610          0253
 0160      NENT(QCVF)=0$ NENT(QUERST)=0$                                     00002611          0254
 0162      NENT(NONOT)=0$ NENT(NOTAB)=0$                                     00002612          0255
 0164      NENT(WORDS)=0$ NENT(STEMS)=0$                                     00002613          0256
 0166      NENT(DELWDS)=0$ NENT(COUNTS)=0$                                   00002614          0257
 0167      NENT(DELCNTS)=0$                                                  00002615          0258
 0167      ''                                                                00002620          0259
 0167      WE ARE GOING TO PROCESS A QUERY AT LAST                           00002630          0260
 0167      ''                                                                00002640          0261
 0167      ''ISOLATE INPUT QUERY TEXT''                                      00002650          0262
 0168      LLINK(6H(ISOLAT))$                                                00002660          0263
 0169      ISOL(=WORDS,COUNTS)$                                              00002670          0264
 0169      ''ANALYZE THE INPUT TEXT                                          00002680          0265
 0169      ''                                                                00002690          0266
 0170      LLINK(6H(ANALYZ))$                                                00002700          0267
 0171      ANAL(WORDS,COUNTS=WORDS,STEMS)$                                   00002710          0268
 0171      ''LOOK UP THE STEMS JUST FOUND                                    00002720          0269
 0171      ''                                                                00002730          0270
 0172      LLINK(6H(LOOKUP))$                                                00002740          0271
 0173      LOOK(WORDS,STEMS,CONTENTS,COUNTS=NOTAB,QUERST,NULQRY)$            00002750          0272
 0173      ''                                                                00002760          0273
 0173      SEE IF RETRIEVAL IS POSSIBLE (QUERY NOT NULL)''                   00002770          0274
 0173                                                                        00002780          0275
 0174      IF NULQRY EQ V(ON)$                                               00002790          0276
 0175        GOTO DIA5$                                                      00002800          0277
 0175      ''                                                                00002810          0278
 0175      ELSE, BUILD THE QUERY CONCEPT VECTOR                              00002820          0279
 0175      ''                                                                00002830          0280
 0176      DIA2'1. IF DOCDOC EQ V(ON) $                                      00002831          0281
 0177         GOTO DIA3 $                                                    00002832          0282
 0178       QCVBLD(QUERST=QCV)$                                              00002840          0283
 0178      ''                                                                00002850          0284
 0178      CHECK MODE-3 FLAG                                                 00002860          0285
 0178      ''                                                                00002870          0286
 0179      IF PRINTQ EQ V(ON)$                                               00002880          0287
 0180        GOTO DIA10$                                                     00002890          0288
 0181      DIA3, HITS=0$                                                     00002900          0289
 0181      ''                                                                00002910          0290
 0181      GO PERFORM A RETRIEVAL BASED ON THE CURRENT QUERY CONCEPT V.      00002920          0291
 0181      ''                                                                00002930          0292
 0182      LLINK(6H(RETRIE))$                                                00002940          0293
 0183      RETRIV(QCV,NONOT,NEWTID=                                          00002950          0294
 0183             TEMP,HITS)$                                                00002951          0295
 0184      BINBCD(NENT(TEMP)=TEMPH)$                                         00002952          0296
 0185      SEND(18H(        TEMP. COUNT),3,18)$                              00002954          0297
 0186      SEND(TEMPH,1,6)$                                                  00002955          0298
 0187      IF HITS GQ 64 $                                                   00002960          0299
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46
                                                                              00002961          0300
 0188      BEGIN                                                              00002962          0301
 0189       MESAGE(19)$                                                       00002963          0302
 0190      GOTO CHAT$                                                         00002964          0303
 0191      END                                                                00002970          0304
 0192     SINBCD(HITS=TEMPH)$''CONVERT''                                      00002971          0305
 0193     MESAGE(20)$                                                         00002972          0306
 0194     SEND(TEMPH,1,6)$                                                    00002990          0307
 0195     IF HITS LQ 0$                                                       00003000          0308
 0196      GOTO DIA4$                                                         00003010          0309
 0197      GOTO CHAT$                                                         00003020          0310
 0198     DIA4.                                                               00003030          0311
 0198     ''                                                                  00003040          0312
 0198     CHECK RPLG                                                          00003050          0313
 0198     ''                                                                  00003060          0314
 0198     IF RPLG EQ V(ON)$                                                   00003070          0315
 0199      GOTO OPT$                                                          00003080          0316
 0200     DIA5.                                                               00003090          0317
 0200     MESAGE(12)$                                                         00003100          0318
 0201     SWITCH DIAAA=(DIA1,DIA5'1,OPT)$                                     00003101          0319
 0202     ZORCH=YESNO(XXX)$                                                   00003102          0320
 0203     GOTO DIAAA($ZORCH$)$                                                00003110          0321
 0204     DIA5'1.                                                             00003120          0322
 0204     MESAGE(13)$                                                         00003130          0323
 0205     SWITCH DIAB=(DIA7,DIA6,OPT)$                                        00003131          0324
 0206     ZORCH=YESNO(XXX)$                                                   00003132          0325
 0207     GOTO DIAB($ZORCH$) $                                                00003140          0326
 0208     DIA6.                                                               00003150          0327
 0209     NEWQ=V(ON)$ GOTO CHG2$                                              00003160          0328
 0210     DIA7. GOTO JOBEND$                                                  00003170          0329
 0211     DIA8. MESAGE(8)$                                                    00003180          0330
 0212     SWITCH DIAC=(OPT,CHAT,OPT)$                                         00003181          0331
 0213     ZORCH=YESNO(XXX)$                                                   00003182          0332
 0214     GOTO DIAC($ZORCH$) $                                                00003190          0333
 0215     DIA10. MESAGE(15)$ ''ASK ABOUT SEEING WORDS AND STEMS''             00003191          0334
 0216     SWITCH DIAD=(,DIA3,OPT)$                                            00003192          0335
 0218     ZILCH=YESNO(XYZ)$ GOTO DIAD($ZILCH$) $                              00003193          0336
 0219     MESAGE(16)$''MAKE HEADER''                                          00003194          0337
 0220     FOR A=ALL(WORDS)$                                                   00003195          0338
 0222      BEGIN BYTE($0,18$)(WRDSTEM)=WRDENT($A$) $                          00003196          0339
 0224     BYTE($20,12$)(WRDSTEM)=STMENT($A$) $ BYTE($18,2$)(WRDSTEM)=         00003197          0340
 0225      2H(  )$ BYTE($32,2$)(WRDSTEM)=BYTE($4,2$)(LINEND)$                 00003198          0341
 0226       SEND(WRDSTEM,6,32)$                                               00003199          0342
 0227      END                                                                00003200          0343
 0228     MESAGE(6$)$''SEE IF "USER" WANTS TO RETRIEVE''                      00003201          0344
 0229     SWITCH DIAE=(DIA3,DIA5,OPT)$                                        00003202          0345
 0231     ZILCH=YESNO(XYZ)$ GOTO DIAE($ZILCH$) $                              00003240          0346
 0232     CHAT. ''"CHAT" FUNCTION BEGINS HERE''                               00003250          0347
 0232     IF PRINTR NQ V(ON)$                                                 00003260          0348
 0233      GOTO CHAT2$                                                        00003270          0349
 0234     CHAT1.
```

Figure VI.28. (Continued)



```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0234      MESAGE(21)$                                            00003280         0350
  0235      SWITCH CHATA=(,CHAT2,OPT)$                             00003290         0351
  0236      ZORCH=YESNO(XYZ)$                                      00003300         0352
  0237      GOTO CHATA($ZORCH$)$                                   00003310         0353
  0238      MESAGE(22)$                                            00003320         0354
  0238      ''NOW..                                                00003330         0355
  0238      FORMAT AND SEND 'TEMP' DATA                            00003340         0356
  0238      ''                                                     00003350         0357
  0239      IF NENT(TEMP) EQ 0$                                    00003360         0358
  0240       GOTO CHAT2$''CAN'T PRINT EMPTY TABLE''                00003370         0359
  0241      TBASE=0$''FOR LIMITED "TEMP" PRINTING''                00003371         0360
  0242      FOR A=ALL(TEMP)$                                       00003380         0361
  0243       BEGIN                                                 00003390         0362
  0244        IF ACCNUM($A$) EQ 0$                                 00003400         0363
  0245         TEST A$                                             00003410         0364
  0246      FOR T=0,6,60$                                          00003420         0365
  0247       BEGIN                                                 00003421         0366
  0248        BYTE($T,6$) (TEMPLINE)=6H(      )$                   00003422         0367
  0249       END                                                   00003430         0368
  0249      ''NOW PUT IN THE DATA''                                00003440         0369
  0250      BINBCD(ACCNUM($A$)=TEMPH)$                             00003441         0370
  0251      BYTE($2,6$)(TEMPLINE)=TEMPH$                           00003450         0371
  0252      BINBCD(TEMPID($A$)=TEMPH)$                             00003460         0372
  0253      BYTE($15,3$)(TEMPLINE)=TEMPH$                          00003461         0373
  0254      BINBCD(RETRNK($A$)=TEMPH)$                             00003470         0374
  0255      BYTE($39,6$)(TEMPLINE)=TEMPH$                          00003471         0375
  0256      BINBCD(BIBSUP($A$)=CHATH)$                             00003480         0376
  0257      BYTE($53$)(TEMPLINE)=BYTE($5$)(CHATH)$                 00003481         0377
  0258      BYTE($57$)  (TEMPLINE)=1H(/) $                         00003490         0378
  0259      BINBCD(LASRET($A$)=CHATH)$                             00003491         0379
  0260      BYTE($61$)(TEMPLINE)=BYTE($5$)(CHATH)$                 00003492         0380
  0260                                                             00003493         0381
  0261      ICOR=(CORCOE($A$) + 5.0E-6)*1.0E3 $                    00003500         0382
  0261                                                             00003501         0383
  0262      BINBCD(ICOR=HICOR)$                                    00003510         0384
  0263      BYTE($27$)  (TEMPLINE)=BYTE($2$) (HICOR)$              00003511         0385
  0264      BYTE($28$)  (TEMPLINE)=1H(,)$                          00003512         0386
  0265      BYTE($29,3$)(TEMPLINE)=BYTE($3,3$) (HICOR)$            00003513         0387
  0265                                                             00003514         0388
  0266      SEND(TEMPLINE,11,62)$                                  00003520         0389
  0267      IF TBASE+4 LS A $                                      00003521         0390
  0268       BEGIN                                                 00003522         0391
  0270        TBASE=A$ MESAGE(3) $                                 00003523         0392
  0271        SWITCH CHATT=(,CHAT2,OPT)$                           00003524         0393
  0272        ZORCH=YESNO(XYZ)$                                    00003525         0394
  0273        GOTO CHATT($ZORCH$)  $                               00003526         0395
  0274       END                                                   00003527         0396
  0275       END                                                   00003530         0397
  0276       MESAGE(25) $ ''THAT'S "ALL" OF THE "TEMP" TABLE''     00003531         0398
  0277      CHAT2.                                                 00003540         0399
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0277        MESAGE(23)$                                              00003550        0400
 0278        SWITCH CHATB=(,OPT,OPT)$                                 00003560        0401
 0279        ZORCH=YESNO(XYZ)$                                        00003570        0402
 0280        GOTO CHATB($ZORCH$)$                                     00003580        0403
 0281        CHAT3.                                                   00003590        0404
 0281        JJ=0$                                                    00003600        0405
 0282        CHAT5.                                                   00003630        0406
 0283         IF ACCNUM($JJ$) LQ 0 $ GOTO CHAT8$                      00003631        0407
 0284        BINBCD(ACCNUM($JJ$)=DATSTF($0$))$                        00003660        0408
 0285        BINBCD(TEMPID($JJ$)=DATSTF($1$))$                        00003665        0409
 0286        BINBCD(RETRNA($JJ$)=DATSTF($2$))$                        00003670        0410
 0287        BINBCD(BIBSUP($JJ$)=DATSTF($3$))$                        00003671        0411
 0288        BINBCD(LASRET($JJ$)=DATSTF($4$))$                        00003672        0412
 0288                                                                 00003673        0413
 0289        ICOR=(CORCOE($JJ$) + 5.0E-6)*1.0E3 $                     00003680        0414
 0290         BINBCD(ICOR=HICOR)$''CONVERT TO BCD FORM''              00003681        0415
 0291        BYTE($1$)(DATSTF($5$))=BYTE($2$)(HICOR)$                 00003682        0416
 0292        BYTE($3,3$)(DATSTF($5$))=BYTE($3,3$)(HICOR) $            00003690        0417
 0293        FOR T=ALL(TMPLN)$                                        00003700        0418
 0294         SEND(TMPFAKE($T$),2,12)$                                00003701        0419
 0297        IF BIBSUP($JJ$) EQ 1 $ BEGIN MESAGE(24)$                 00003710        0420
 0299         GOTO CHAT6$ END                                         00003720        0421
 0300         SNDBIB( ACCNUM($JJ$) )$''SEND BIBLIO. INFO''            00003730        0422
 0301        BIBSUP($JJ$)=1$''SET BIBLI SUPPASS BIT''                 00003740        0423
 0302        CHAT6.                                                   00003750        0424
 0302        JJ=JJ+1$                                                 00003760        0425
 0303        IF JJ GQ NENT(TEMP)$                                     00003770        0426
 0304         GOTO CHAT8$                                             00003780        0427
 0305        CHAT7. MESAGE(3)$ ''MORE OF THE SAME''                   00003820        0428
 0306        SWITCH CHATC=(CHAT5,OPT,OPT)$                            00003830        0429
 0307        ZORCH=YESNO(XYZ)$                                        00003840        0430
 0308        GOTO CHATC($ZORCH$)$                                     00003850        0431
 0309        CHAT8. MESAGE(25)$                                       00003860        0432
 0309                                                                 00003880        0433
 0310        OPT.''OPTIONS''                                          00003900        0434
 0311        OPT1. MESAGE(27)$ PLACE=6H(      )$                      00003910        0435
 0312        PL=1$                                                    00003915        0436
 0313        RECV(=PLACE,PL,CCH)$                                     00003916        0437
 0314        OPT7.                                                    00003917        0438
 0314        IF BYTE($0,4$) (PLACE) EQ 4H(HELP)$                      00003920        0439
 0315         GOTO OPT2$                                              00003930        0440
 0316        FOR X=ALL(OPTNAM)$                                       00003940        0441
 0317         BEGIN                                                   00003950        0442
 0320          IF PLACE EQ NAMES($X$)$BEGIN ZILCH=X$                  00003960        0443
 0321           GOTO OPT1'1$                                          00003970        0444
 0322           END                                                   00003971        0445
 0323           TEST X$                                               00003980        0446
 0324         END                                                     00003990        0447
 0325        GOTO OPT6$                                               00004000        0448
 0326        OPT1'1. DEFLG=V(OFF)$                                    00004010        0449
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0327      FFLG=V(OFF)$                                                 00004020        0450
 0328      GOTO OPTSW($ZILCH$)$                                         00004030        0451
 0329      OPT2.                                                        00004040        0452
 0329      MESAGE(28)$                                                  00004050        0453
 0330      IF OPTION EQ V(ON)$                                          00004060        0454
 0331       GOTO OPT3$                                                  00004070        0455
 0332       GOTO OPT4$                                                  00004080        0456
 0333      OPT3, MESAGE(29)$                                            00004090        0457
 0334      GOTO OPT1$                                                   00004100        0458
 0335      OPT4,                                                        00004110        0459
 0335      MESAGE(30)$                                                  00004120        0460
 0336      SWITCH OPTA=(OPT3,OPT1,OPT1)$                                00004130        0461
 0337      ZORCH=YESNO(XYZ)$                                            00004140        0462
 0338      GOTO OPTA($ZORCH$)$                                          00004150        0463
 0339      OPT6, MESAGE(31)$                                            00004160        0464
 0340      GOTO OPT1$                                                   00004170        0465
 0340                                                                   00004180        0466
 0341      CHG.''CHANGE-MODES FUNCTION''                                00004190        0467
 0341                                                                   00004200        0468
 0341      NEWQ=V(OFF)$                                                 00004210        0469
 0342      CHG2, FOR A=1,1,5$                                           00004220        0470
 0343       MODEF($A$)=V(OFF)$                                          00004230        0471
 0344      CHGCHR=0$''CLEAR OUT''                                       00004231        0472
 0345      MESAGE(34)$                                                  00004240        0473
 0346      SWITCH CHGA=(,CHG3,OPT)$                                     00004250        0474
 0347      ZORCH=YESNO(XYZ)$                                            00004260        0475
 0348      GOTO CHGA($ZORCH$)$                                          00004270        0476
 0349      MESAGE(35)$                                                  00004280        0477
 0350      CHG3, MESAGE(14)$                                            00004290        0478
 0351      PL=2$                                                        00004291        0479
 0352      RECV(=MODESTR,PL,CCN)$                                       00004300        0480
 0353      CHG4, CHGBPOS=0$''BYTE POINTER''                             00004320        0481
 0354       GOTO CHG6 $                                                 00004321        0482
 0355      CHG5, CHGBPOS=CHGBPOS+1$                                     00004330        0483
 0356      CHG6,                                                        00004340        0484
 0356      BYTE($5$) (CHBYTE)=BYTE($CHGBPOS$) (MODESTR)$                00004350        0485
 0357      IF CHGCHR LQ 5 $                                             00004360        0486
 0358       GOTO CHG7$                                                  00004370        0487
 0359      IF CHBYTE EQ 6H(00000,)$                                     00004380        0488
 0360       GOTO CHG5$                                                  00004390        0489
 0361      MESAGE(73)$''ILLEGAL STRING''                                00004400        0490
 0362      IF NEWQ EQ V(ON)$                                            00004410        0491
 0363        GOTO CHG2$                                                 00004420        0492
 0364        GOTO CHG $                                                 00004430        0493
 0365      CHG7, IF CHGCHR EQ 0 $                                       00004440        0494
 0366       BEGIN                                                       00004450        0495
 0367       IF NEWQ EQ V(ON)$                                           00004460        0496
 0368           GOTO DIAO$                                              00004470        0497
 0369           GOTO OPT$                                               00004480        0498
 0370        END                                                        00004490        0499
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08,567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0371      CHG8,                                                        00004500     0500
 0371      IF CHGCHR EQ 1 $''"TERSE" NEEDS COMPLEMENTING''              00004511     0501
 0372       BEGIN                                                       00004512     0502
 0373        IF TERSE EQ V(ON)$'' IF IT'S "ON" ''                       00004513     0503
 0374         BEGIN                                                     00004514     0504
 0375          TERSE=V(OFF)$''TURN IT "OFF"''                           00004515     0505
 0376          MESAGE(-1)$''LOAD VERBOSE MESSAGES''                     00004516     0506
 0377          GOTO CHG5$                                               00004517     0507
 0378         END                                                       00004518     0508
 0379        TERSE=V(ON)$''ELSE, TURN IT "ON"''                         00004519     0509
 0380        MESAGE(0)$''AND LOAD "TERSE" MESSAGES''                    00004520     0510
 0381        GOTO CHG5$                                                 00004521     0511
 0382       END                                                         00004522     0512
 0383      MODEF($CHGCHR-1$)=V(ON)$''TURN "ON" INDICATED FLAG''         00004523     0513
 0384      GOTO CHG5$                                                   00004524     0514
 0384                                                                   00004530     0515
 0385      DDC,''DOCUMENT-DOCUMENT CORRELATION FUNCTION''               00004540     0516
 0385      LLINK(6H(NUMBAN))$                                           00004541     0517
 0385      ''CLEAR'WORDS' ''                                            00004550     0518
 0386      NENT(WORDS)=0$                                               00004580     0519
 0387      IF DOCDOC NQ V(ON)$                                          00004590     0520
 0388        GOTO DDC1$                                                 00004600     0521
 0389      MESAGE(54)$                                                  00004610     0522
 0390      SWITCH DDCA=(DDC2,DDC1,OPT)$                                 00004620     0523
 0391      SORCH=YESNO(XYZ)$                                            00004630     0524
 0392      GOTO DDCA($SORCH$)$                                          00004640     0525
 0393      DDC1,                                                        00004650     0526
 0393      FOR Q=ALL(QCV)$                                              00004660     0527
 0394       BEGIN                                                       00004670     0528
 0395        CORREL($Q$)=0,C$                                           00004690     0529
 0396       END                                                         00004700     0530
 0397      DDC2,                                                        00004720     0531
 0397      RESET=0$                                                     00004721     0532
 0398      MESAGE(52)$                                                  00004730     0533
 0399      DOCDOC=V(ON)$                                                00004740     0534
 0400      DDC3, NUMBER(TEMP,RESET=RESET,DOCUNUM,ACORTM)$               00004750     0535
 0401      IF DOCUNUM LS 0 $                                            00004760     0536
 0402        GOTO MOD11$                                                00004770     0537
 0403      IF ACORTM EQ V(TM) $                                         00004780     0538
 0404       DOCUNUM=ACCNUM($DOCUNUM$) $''FOR TRUE ACCESSION # ''        00004790     0539
 0405      IF DOCUNUM EQ 0 $                                            00004800     0540
 0406       GOTO DDC3 $ ''NOT REALLY IN THE TABLE''                     00004810     0541
 0407      FETDCV(DOCUNUM=DDDDDD) $ ''FETCH THE DCV''                   00004820     0542
 0408      IF DCVCON($0$) NQ DOCUNUM $ ''DOUBLE CHECK IT''              00004830     0543
 0409       BEGIN                                                       00004840     0544
 0410        SEND(12H(FETDCV FAIL=),2,12) $                             00004850     0545
 0411        BINBCD(DOCUNUM=TEMPH) $                                    00004860     0546
 0412        SEND(TEMPH,1,6) $                                          00004870     0547
 0413        GOTO DDC3 $                                                00004880     0548
 0414       END                                                         00004890     0549
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0415       FOR W=1,1,256$ ''PERFORM QCV-BLD WITH DCV ''                          00004900          0550
  0416        BEGIN                                                                00004901          0551
  0417         IF DCVCON($WS) EQ 0 $                                               00004902          0552
  0420          BEGIN GOTO MOD11$ END                                              00004903          0553
  0421        CORREL($DCVCON($WS)$)=CORREL($DCVCON($WS)$)+DCVWGT($WS) $            00004910          0554
  0422        END                                                                  00004915          0555
  0423       GOTO MOD11 $ ''EXIT TO "MOD"-11 ''                                    00004920          0556
  0423                                                                             00004950          0557
  0423       ''DOC FUNCTION''                                                      00004960          0558
  0424       DOC. LLINK(6H(NUMBAH))$                                               00004970          0559
  0425        IF NENT(TEMP) EQ 0 $                                                 00004971          0560
  0426         GOTO DOC1$                                                          00004980          0561
  0427       MESAGE(47)$                                                           00004990          0562
  0428       SWITCH DOCA=(CHAT1,DOC1,OPT)$                                         00005000          0563
  0429       ZILCH=YESNO(XXX)$                                                     00005010          0564
  0430       GOTO DOCA($ZILCH$)$                                                   00005020          0565
  0432       DOC1, MESAGE(68)$ RESET=0$                                            00005030          0566
  0434       DOC2, NUMBER(TEMP,RESET=RESET,NEXTSFQ,ACCRTM)$ TMPOINT=-1 $           00005040          0567
  0435        IF NEXTSEQ EQ -1 $                                                   00005050          0568
  0436         GOTO OPT$                                                           00005060          0569
  0437       IF ACORTM EQ V(AC)$                                                   00005070          0570
  0438         BEGIN''IT WAS AN ACCESSION NUMBER''                                 00005080          0571
  0438                                                                             00005081          0572
  0438                                                                             00005091          0573
  0439          FOR A=ALL(TEMP)$                                                   00005100          0574
  0440           BEGIN                                                             00005110          0575
  0441            IF NEXTSEQ EQ ACCNUM($A$)$                                       00005120          0576
  0442             BEGIN                                                           00005130          0577
  0443              TMPOINT=A$                                                     00005140          0578
  0444              GOTO DOC3$                                                     00005150          0579
  0445            END                                                              00005160          0580
  0446          END                                                                00005170          0581
  0447       GOTO DOC4$''NOT IN TEMP TABLE AT ALL''                                00005180          0582
  0448       END                                                                   00005190          0583
  0448       ''IT WAS A TEMPORARY ID, NUMBER,                                      00005200          0584
  0448         LOOK FOR THE TRUE ACCESSION NUMBER''                                00005210          0585
  0449       TMPOINT=NEXTSEQ$                                                      00005220          0586
  0450       DOC3, ''FORMAT AND SEND TEMP, INFO TO USER''                          00005320          0587
  0450       DATSTF($0$)=12H(             )$                                       00005321          0588
  0451       DATSTF($1$)=12H(             )$                                       00005322          0589
  0452       BINBCD(ACCNUM($TMPOINT$)=NUMB)$                                       00005330          0590
  0453        BYTE($1,4$)(DATSTF($0$))=BYTE($2,4$)(NUMB)$                          00005340          0591
  0453       ''SET UP TEMP ID.''                                                   00005350          0592
  0454       BINBCD(TEMPID($TMPOINT$)=NUMB)$                                       00005360          0593
  0455        BYTE($1,4$)(DATSTF($1$))=BYTE($2,4$)(NUMB)$                          00005370          0594
  0455       ''SET UP RETRIEVAL BANK''                                             00005380          0595
  0456       BINBCD(RETRNK($TMPOINT$)=NUMB)$                                       00005390          0596
  0457        DATSTF($2$)=12H(             )$                                      00005391          0597
  0458        BYTE($1,4$)(DATSTF($2$))=BYTE($2,4$)(NUMB)$                          00005400          0598
  0458                                                                             00005401          0599
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VFRSION -- NUMBER 46

  0459       IF BIBSUP($TMPOINT$) EQ 0 $                                        00005410             0600
  0460      BEGIN                                                               00005420             0601
  0462       DATSTF($3$)=12H(  NO
```

Figure VI.28. (Continued)

```
0463      END                                                   00005440        0603
0464       DATSTP($3$)=12H(   YES
```

Figure VI.28. (Continued)

```
0465      DOC3'1. IF LASRET(STMPOINTS)EQ 0 $                                  00005460          0605
0466      BEGIN                                                               00005470          0606
0468       DATSTF($4$)=12H(   NO
```

Figure VI.28. (Continued)

```
0469      END                                                        00005490        0608
0470       DATSTF($4$)=12H(   YES
```

Figure VI. 28. (Continued)

```
0471     DOC3'2. DOCCORF=CORCOE($TMPOINT$)$''GET CORRELATION''                    00005510     0610
0472     DOCCORF=DOCCORF*1,0E3$''MOVE DECIMAL POINT''                             00005520     0611
0473     DOCCOR=DOCCORF$''FLOAT TO INTEGER''                                      00005530     0612
0473                                                                              00005531     0613
0474     BINBCD(DOCCOR=DOCCORH)$                                                  00005540     0614
0475     BYTE($1$)(DATSTF($5$))=BYTE($2$)(DOCCORH)$''WHOLE PART''                 00005550     0615
0476     BYTE($3,3$)(DATSTF($5$))=BYTE($3,3$)(DOCCORH)$''FRACTIONAL PART''        00005560     0616
0476     ''NOW SEND THE STUFF AWAY TO THE USER''                                  00005570     0617
0476                                                                              00005571     0618
0477     FOR S=ALL(TMPLN)$                                                        00005580     0619
0478      SEND(TMPFAKE($S$),2,12) $                                               00005590     0620
0479     IF BIBSUR($TMPOINT$) EQ 0 $                                              00005600     0621
0480        GOTO DOC4$                                                            00005610     0622
0481     PS=V(ON)$                                                                00005620     0623
0482     GOTO DOC5$                                                               00005630     0624
0483     DOC4, PS=V(OFF)$                                                         00005640     0625
0484     IF NENT(NONOT) NQ 0$                                                     00005650     0626
0485        GOTO DOC6$''NOTHING ON THE "NONO" LIST SO WHY CHECK''               00005660     0627
0485     ''CHECK REQUEST AGAINST "NONO" LIST''                                    00005670     0628
0486     DOC5,                                                                    00005675     0629
0486     FOR N=ALL(NONOT)$                                                        00005680     0630
0487      BEGIN                                                                   00005690     0631
0488       IF ACCNUM($TMPOINT$) EQ NONO($N$)$                                     00005700     0632
0489        BEGIN                                                                 00005710     0633
0490         MESAGE(48)$                                                          00005720     0634
0491         GOTO DOC6 $                                                          00005730     0635
0492        END                                                                   00005740     0636
0493      END                                                                     00005750     0637
0494     DOC6, IF PS EQ V(OFF)$                                                   00005760     0638
0495       GOTO DOC7$                                                             00005770     0639
0496     MESAGE(49)$                                                              00005780     0640
0497     SWITCH DOCB=(,DOC8,OPT)$                                                 00005790     0641
0498     SILCH=YESNO(XXX)$                                                        00005800     0642
0499     GOTO DOCB($SILCH$)$                                                      00005810     0643
0500     DOC7, IF TMPOINT EQ -1$                                                  00005811     0644
0501                 BEGIN                                                        00005812     0645
0502                 SNDSIB(NEXTSEQ) $                                            00005813     0646
0503                 GOTO DOC8$                                                   00005814     0647
0504                 END                                                          00005815     0648
0504                                                                              00005816     0649
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0505      SNDBIB(ACCNUM(STMPOINTS)) $                                    00005820          0650
 0506     DOC8, MESAGE(50)$                                               00005830          0651
 0507     SWITCH DOCC=(,DOC9,OPT)$                                        00005840          0652
 0508     ZILCH=YESNO(XXX)$                                               00005850          0653
 0509     GOTO DOCC($ZILCH$)$                                             00005860          0654
 0510      IF TMPOINT EQ -1 $                                             00005861          0655
 0511       BEGIN                                                         00005862          0656
 0512        SNDABS(NEXTSEQ) $                                            00005863          0657
 0513        GOTO DOC9$                                                   00005864          0658
 0514       END                                                           00005865          0659
 0515     SNDABS(ACCNUM(STMPOINTS))$                                      00005880          0660
 0516     DOC9, MESAGE(33)$                                               00005890          0661
 0517     SWITCH DOCD=(DOC1,OPT,OPT)$                                     00005900          0662
 0518     ZILCH=YESNO(XXX)$                                               00005910          0663
 0519     GOTO DOCD($ZILCH$)$                                             00005920          0664
 0519     ''RET FUNCTION''                                                00005950          0665
 0520     RET.                                                            00005960          0666
 0520     IF NENT(TEMP) NQ 0$                                             00005970          0667
 0521      GOTO RET1$                                                     00005980          0668
 0522     MESAGE(41)$.                                                    00005981          0669
 0523     RET1, MESAGE(43)$                                               00005990          0670
 0524     SWITCH RETA=(,RET1'2,OPT)$                                      00006010          0671
 0525     ZORCH=YESNO(XYZ)$                                               00006020          0672
 0526     GOTO RETA($ZORCH$)$                                             00006030          0673
 0527     NENT(NONOT)=0$                                                  00006031          0674
 0528     FOR T=ALL(TEMP)$                                                00006040          0675
 0529     FOR N=NENT(NONOT),1$''NONO INDEX COUNTER''                      00006050          0676
 0530      BEGIN                                                          00006060          0677
 0531       IF ACCNUM($T$) NQ 0$                                          00006070          0678
 0532        BEGIN                                                        00006080          0679
 0533         NONO($N$)=ACCNUM($T$)$                                      00006090          0680
 0534         ZOT=N$                                                      00006100          0681
 0535         TEST N$                                                     00006110          0682
 0536        END                                                          00006120          0683
 0537        TEST T $                                                     00006130          0684
 0538      END                                                            00006140          0685
 0539     NENT(NONOT)=NENT(NONOT)+ZOT$                                    00006150          0686
 0540     GOTO RET2$                                                      00006160          0687
 0541     RET1'2. NENT(NONOT)=0$''NONO CLEARED LOGICALLY)''               00006170          0688
 0542     RET2,                                                           00006180          0689
 0542     MESAGE(44)$                                                     00006190          0690
 0543     SWITCH RETB=(RET3,,OPT)$                                        00006200          0691
 0544     ZORCH=YESNO(XYZ)$                                               00006210          0692
 0545     GOTO RETB($ZORCH$)$                                             00006220          0693
 0546     FOR A=ALL(TEMP)$                                                00006230          0694
 0547      BIBSUP($A$)=0$                                                 00006240          0695
 0548     RET3, DEFLG=V(OFF)$                                             00006250          0696
 0549     IF NENT(TEMP) LS 256$                                           00006260          0697
 0550      GOTO RET5$                                                     00006270          0698
 0551     MESAGE(46)$                                                     00006280          0699
```

Figure VI.28. (Continued)

```
0023T 01  05-14-71   08.367   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0552      RET4, DEFLG=V(ON)$                                                          00006290          0700
  0553      GOTO DEL$                                                                   00006300          0701
  0554      RET5, IF NENT(TEMP) EQ 0 $                                                  00006310          0702
  0557       BEGIN MESAGE(41)$ MESAGE(42)$                                              00006311          0703
  0558             SEND(6H(     1),1,6)$                                                00006312          0704
  0560             NEWTID=1$ GOTO DIA2'1 $                                              00006313          0705
  0561       END                                                                        00006314          0706
  0562      COMTMP(TEMP)$''COMPRESS "TEMP" ''                                           00006315          0707
  0563      SMALL=0$                                                                    00006316          0708
  0564      FOR T=ALL(TEMP)$                                                            00006317          0709
  0565       BEGIN                                                                      00006318          0710
  0566        IF TEMPID($T$) GR SMALL $                                                 00006319          0711
  0567         SMALL=TEMPID($T$)  $                                                     00006320          0712
  0568       END                                                                        00006321          0713
  0569      NEWTID=SMALL+1$                                                             00006322          0714
  0570      BINBCD(NEWTID=HICOR)$                                                       00006323          0715
  0572      SEND(HICOR,1,6)$ MESAGE(42)$                                                00006324          0716
  0573      BINBCD(256-NENT(TEMP)=HICOR)$                                               00006325          0717
  0574      SEND(HICOR,1,6)$                                                            00006326          0718
  0575      MESAGE(45)$                                                                 00006327          0719
  0576      SWITCH RETC=(RET4,DIA2'1,OPT)$                                              00006328          0720
  0577      SORCH=YESNO(XXX)$                                                           00006329          0721
  0578      GOTO RETC($SORCH$) $                                                        00006330          0722
  0578                                                                                  00006400          0723
  0579      WRD.''WORD FUNCTION, AS WELL AS SEE''                                       00006410          0724
  0579      WRDFLG=V(ON)$                                                               00006420          0725
  0580      GOTO WRD2$                                                                  00006430          0726
  0581      SEE.                                                                        00006440          0727
  0581      SEEFLG=V(ON)$                                                               00006450          0728
  0582      WRD2, IF NENT(WORDS) LQ 0 $                                                 00006460          0729
  0583       BEGIN                                                                      00006461          0730
  0584        MESAGE(11)$                                                               00006462          0731
  0585        GOTO OPT$                                                                 00006463          0732
  0586       END                                                                        00006464          0733
  0587      MESAGE(54)$                                                                 00006470          0734
  0588      FOR A=ALL(WORDS)$                                                           00006480          0735
  0590       BEGIN BYTE($0,COUNTER($A$)$)(WRDSTEM)=BYTE($0,CCUNTER($A$)$)               00006490          0736
  0591       (WRDENT($A$))$ BYTE($COUNTER($A$),20)(WRDSTEM)=BYTE($4,20)(LINEND)$        00006491          0737
  0592       SEND(WRDSTEM,4,20)$                                                        00006492          0738
  0593      END                                                                         00006493          0739
  0594      IF SEEFLG EQ V(ON)$                                                         00006500          0740
  0595       GOTO WGT$                                                                  00006510          0741
  0596       GOTO MOD4$                                                                 00006520          0742
  0596                                                                                  00006530          0743
  0597      CLR.''CLEAR FUNCTION''                                                      00006540          0744
  0597      FOR Y=ALL(QCV)$                                                             00006570          0745
  0598       BEGIN                                                                      00006580          0746
  0599        CORREL($Y$)=0,0$                                                          00006600          0747
  0600       END                                                                        00006610          0748
  0601      DOCDOC=V(OFF)$                                                              00006620          0749
```

Figure VI.28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0602      WENT(DELWDS)=0$                                          00006630      0750
 0603      GOTO OPT$                                                00006650      0751
 0603                                                               00006660      0752
 0603      ''DEL FUNCTION''                                         00006670      0753
 0604      DEL. RESET=0$                                            00006680      0754
 0605       LLINK(6H(NUMBAH))$                                      00006681      0755
 0606      MESAGE(36)$                                              00006690      0756
 0607      IF DEFLG EQ V(OFF)$                                      00006700      0757
 0608       GOTO DEL1$                                              00006710      0758
 0609      MESAGE(37)$                                              00006720      0759
 0610      DEL1.                                                    00006730      0760
 0610      MESAGE(38)$                                              00006740      0761
 0611      DEL2.                                                    00006750      0762
 0611      NUMBER(TEMP,RESET=RESET,DOCUNUM,ACORTM)$                 00006760      0763
 0612      IF DOCUNUM LS 0 $                                        00006770      0764
 0613       GOTO DEL4$                                              00006771      0765
 0614       GOTO DEL3$                                              00006780      0766
 0615      DEL4. CONTMP(=TEM") $''COMPRESS THE "TEMP" TABLE''       00006810      0767
 0616       IF DEFLG EQ V(ON) $                                     00006811      0768
 0617       GOTO RET3$                                              00006820      0769
 0618       GOTO OPT$                                               00006830      0770
 0619      DEL3. IF ACORTM EQ V(TM)$                                00006840      0771
 0620          BEGIN ''REMOVE ENTRY''                               00006890      0772
 0621           ENTRY(TEMP($DOCUNUM$))=0$                           00006900      0773
 0622           CORCOE($DOCUNUM$)=0.0$                              00006910      0774
 0623           GOTO DEL4$                                          00006920      0775
 0624          END                                                  00006930      0776
 0625      FOR T=ALL(TEMP)$                                         00006960      0777
 0626         BEGIN                                                 00006970      0778
 0627           IF DOCUNUM EQ ACCNUM($T$) $                         00006971      0779
 0628            BEGIN                                              00006972      0780
 0628                                                               00006973      0781
 0629        ENT(TEMP($T$))=0$                                      00006980      0782
 0630          CORCOE($T$)=0.0$                                     00006990      0783
 0631          GOTO DEL4$                                           00007000      0784
 0632         END                                                   00007001      0785
 0633         END                                                   00007010      0786
 0634      GOTO DEL4$                                               00007020      0787
 0634                                                               00007040      0788
 0635      MOD.''MODIFY FUNCTION''                                  00007050      0789
 0635      WFLG=V(ON)$                                              00007060      0790
 0636      IF DOCDOC EQ V(ON)$                                      00007070      0791
 0637       BEGIN                                                   00007080      0792
 0638        MESAGE(69)$                                            00007090      0793
 0639        GOTO MOD1$                                             00007100      0794
 0640       END                                                     00007110      0795
 0641      MESAGE(51)$                                              00007120      0796
 0642      SWITCH MODA=(DDC,MOD1'1,OPT)$                            00007130      0797
 0643      MOD1. ZOECH=YESNO(XYZ)$                                  00007140      0798
 0644      GOTO MODA($ZOECH$)$                                      00007150      0799
```

Figure VI.28. (Continued)

```
0023T 01  05-14-71   08,567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0645       MOD1'1, MESAGE(70)$                                               00007160        0800
  0646       SWITCH MODB=(MOD2,MOD9,OPT)$                                      00007170        0801
  0647       ZORCH=YESNO(XYZ)$                                                 00007180        0802
  0648       GOTO MODB($ZORCH$)$                                               00007190        0803
  0651       MOD2, NENT(WORDS)=0$ NENT(STEMS)=0$ NENT(COUNTS)=0$               00007200        0804
  0654       MOD3, NENT(DELWDS)=0$ NENT(DELCNTS)=0$ NENT(NOTAB)=0$             00007220        0805
  0655       DOCDOC=V(OFF)$                                                    00007230        0806
  0656       WFLG  =V(ON)$                                                     00007240        0807
  0657       MESAGE(56)$                                                       00007250        0808
  0658       LLINK(6H(ISOLAT))$                                                00007260        0809
  0659       ISOL(=WO.DS,COUNTS)$                                              00007270        0810
  0660       LLINK(6H(ANALYZ))$                                                00007280        0811
  0661       ANAL(WORDS,COUNTS=WORDS,STEMS)$                                   00007290        0812
  0662       LLINK(6H(LOOKUP))$                                                00007300        0813
  0663       LOOK(WORDS,STEMS,CONTENTS,COUNTS=NOTAB,QUERST,NUIQRY)$            00007310        0814
  0664       IF NULQRY EQ V(OFF) $                                             00007320        0815
  0665        GOTO MOD6$                                                       00007330        0816
  0666       MOD4, MESAGE(55)$                                                 00007340        0817
  0667       SWITCH MODBB=(MOD3,MOD6,OPT)$                                     00007350        0818
  0668       MOD5, ZORCH=YESNO(XXX)$                                           00007351        0819
  0669       GOTO MODBB($ZORCH$)$                                              00007352        0820
  0670       MOD6, MESAGE(57)$                                                 00007360        0821
  0671       SWITCH MODC=(,MOD1'1,OPT)$                                        00007370        0822
  0672       ZORCH=YESNO(XYZ)$                                                 00007380        0823
  0673       GOTO MODC($ZORCH$)$                                               00007390        0824
  0674       DOCDOC=V(OFF)$                                                    00007400        0825
  0675       WFLG=V(ON)$                                                       00007410        0826
  0675       ''                                                                00007420        0827
  0675       NOW BUILD TABLE OF WORDS TO DELETE                                00007430        0828
  0675       ''                                                                00007440        0829
  0676       MESAGE(56)$                                                       00007441        0830
  0677       LLINK(6H(ISOLAT))$                                                00007450        0831
  0680       NENT(DELWDS)=0$ NENT(NOTAB)=0$  NENT(DELCNTS)=0$                  00007460        0832
  0681       ISOL(=DELWDS,DELCNTS)$                                            00007461        0833
  0682       FOR C=ALL(DELWDS)$                                                00007470        0834
  0684        BEGIN ''C'' FOR D=ALL(WORDS) $                                   00007480        0835
  0685         BEGIN ''D''                                                     00007490        0836
  0686          IF DELWD($C$) EQ WRDENT($D$) $                                 00007500        0837
  0688           BEGIN WRDENT($D$)=18H(                    )$                  00007510        0838
  0689                 STMENT($D$)=12H(            ) $                         00007511        0839
  0690                 COUNTER($D$)=0$                                         00007520        0840
  0691                 TEST C $                                                00007530        0841
  0692           END                                                           00007540        0842
  0693         END ''D''                                                       00007541        0843
  0694          NNOT($NENT(NOTAB)$)=DELWD($C$) $                               00007550        0844
  0695          NENT(NOTAB)=NENT(NOTAB)+1$                                     00007560        0845
  0696        END ''C''                                                        00007580        0846
  0696       ''SEARCH FINISHED FOR WORDS TO BE DELETED ,,                      00007581        0847
  0696         CONTINUE ONWARDS ''                                             00007590        0848
  0696       ''NOW CHECK IF ALL WORDS COULD INDEED BE REMOVED                  00007600        0849
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMB.R 46
  0696         FROM 'WORDS' TABLE''                                               00007610     0850
  0697        IF NENT(NOTAB) EQ 0 $                                               00007620     0851
  0698         GOTO MOD8$''YES NO NONEXISTENT WORDS TRIED''                       00007630     0852
  0698        ''SORRY, YOU GOOFED''                                               00007640     0853
  0699        MOD7.                                                               00007650     0854
  0699        MESAGE(58)$                                                         00007660     0855
  0700        FOR A=ALL(NOTAB)$''SEND THE OFFENDING WORDS TO THE USER''           00007670     0856
  0704        BEGIN ''A'' FOR B=0,1,16$ BEGIN ''B'' IF BYTE($B$)(NNOT($A$))       00007680     0857
  0706         EQ 1H( ) $ BEGIN BYTE($B,2$)(WRDSTEM)=BYTE($4,2$)(LINEND)$         00007681     0858
  0709           SEND(WRDSTEM,3,18)$ TEST A $ END                                 00007682     0859
  0710         BYTE($B$)(WRDSTEM)=BYTE($B$)(NNOT($A$)) $                          00007683     0860
  0711         END ''B''                                                          00007684     0861
  0712        SEND(NNOT($A$),3,18) $                                              00007685     0862
  0713        END ''A'' ''"TRIMMED" BAD WORDS SENT ''                             00007686     0863
  0714        NENT(NOTAB)=0$''RESET "BAD" COUNT''                                 00007860     0864
  0715        MOD8, LLINK(6H(ANALYZ))$''LOAD ANALYZER''                           00007880     0865
  0716         ANAL(WORDS,COUNTS=WORDS,STEMS)$                                    00007881     0866
  0717         LLINK(6H(LOOKUP))$''LOAD DICT, LOOKUP''                            00007882     0867
  0718         LOOK(WORDS,STEMS,CONTENTS,COUNT"=NOTAB,CUERST,NULQRY)$             00007883     0868
  0719         QCVBL"(QUERST=QCV)$                                                00007884     0869
  0720        IF WRDFLG EQ V(OFF)$                                                00007890     0870
  0721         GOTO MOD1'1$                                                       00007900     0871
  0722        WRDFLG=V(OFF)$                                                      00007910     0872
  0723        GOTO OPT$                                                           00007920     0873
  0724        MOD9.                                                               00007930     0874
  0724        MESAGE(53)$                                                         00007940     0875
  0725        SWITCH MODD=(WRD2,MOD10,OPT)$                                       00007950     0876
  0726        ZORCH=YESNO(XYZ)$                                                   00007960     0877
  0727        GOTO MODD($ZORCH$)$                                                 00007970     0878
  0728        MOD10.                                                              00007980     0879
  0728        IF WFLG EQ V(OFF)$                                                  00007990     089C
  0729         GOTO MOD11$                                                        00008000     0881
  0729        ''CLEAR THE QCV''                                                   00008010     0882
  0730        FOR C=ALL(QCV) $                                                    00008020     0883
  0731          CORREL($C$)=0.0$                                                  00008050     0884
  0732        LLINK(6H(ANALYZ))$                                                  00008060     0885
  0733        ANAL(WORDS,COUNTS=WORDS,STEMS)$                                     00008061     0886
  0734         LLI"K(6H(LOOKUP))$                                                 00008062     0887
  0735        LOOK(WORDS,STEMS,CONTENTS,COUNTS=NOTAB,QUERST,NULQRY)$              00008063     0888
  0736        QCVBLD(QUERST=QCV)$                                                 00008070     0889
  0737        MOD11.                                                              00008080     0890
  0737        MESAGE(60)$                                                         00008090     0891
  0738        SWITCH MOD'E=(WGT2,MOD12,OPT)$                                      00008100     0892
  0739        ZORCH=YESNO(XYZ)$                                                   00008110     0893
  0740        GOTO MOD'E($ZORCH$)$                                                00008120     0894
  0741        MOD12.                                                              00008130     0895
  0741        MESAGE(65)$                                                         00008140     0896
  0742        SWITCH MODF=(,MOD,OPT)$                                             0000815C     0897
  0743        ZORCH=YESNO(XYZ)$                                                   00008160     0898
  0744        GOTO MODF($ZORCH$)$                                                 00008170     0899
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08,567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0745      FOR Q=ALL(QCV)$                                                00008180        0900
  0746       BEGIN ''Q''                                                   00008190        0901
  0747        IF CORREL($Q$) NQ 0.0 $                                      00008200        0902
  0748         GOTO RET$''JUMP OUT ON FIRST NON-ZERO CORRELATION''         00008201        0903
  0749       END ''Q''                                                     00008202        0904
  0750      MOD13,                                                         00008210        0905
  0750      MESAGE(66)$                                                    00008220        0906
  0751      SWITCH MODG=(ENDF,OPT,OPT)$                                    00008230        0907
  0752      ZORCH=YESNO(XYZ)$                                              00008240        0908
  0753      GOTO MODG($ZORCH$)$                                            00008250        0909
  0753                                                                     00008260        0910
  0753      ''CON FUNCTION:                                                00008270        0911
  0753        EXAMINES DCV'S FOR                                           00008280        0912
  0753        SELECTED ACCESSION NUMBERS                                   00008290        0913
  0753        IN THE DOCUMENT COLLECTION''                                 00008300        0914
  0753                                                                     00008310        0915
  0754      CON. RESET=0$                                                  00008320        0916
  0755      MESAGE(52)$                                                    00008321        0917
  0756       LLINK(6H(NUMBAN)) $                                           00008322        0918
  0757      CON1, NUMBER(TEMP,RESET=RESET,DOCUNO,ACORTM)$                  00008330        0919
  0758      IF DOCUNO LS 0$                                                00008340        0920
  0759       GOTO CON3$                                                    00008350        0921
  0760      MESAGE(39)$''ACCESSION # IS:''                                 00008380        0922
  0760      ''NOW LOCATE THE TRUE ACCESSION NUMBER, IF REF.                00008390        0923
  0760        WAS MADE TO TEMP TABLE BY THE USER''                         00008400        0924
  0761      IF ACORTM EQ V(TM)$                                            00008410        0925
  0762       BEGIN                                                         00008420        0926
  0763        IF NENT(TEMP) EQ 0$                                          00008430        0927
  0763      ''SORRY, YOU B O O B,  NOTHING IN TEMP TABLE AT ALL''          00008440        0928
  0764         BEGIN                                                       00008450        0929
  0765          MESAGE(5)$                                                 00008460        0930
  0766          GOTO CON$                                                  00008470        0931
  0767         END                                                         00008480        0932
  0768      DOCUNO=ACCNUM($DOCUNO$) $                                      00008490        0933
  0769         END                                                         00008570        0934
  0770      CON2,''DOCUNO=A TRUE ACCESSION NUMBER''                        00008580        0935
  0770      FETDCV(DOCUNO=DDDDDD)$''GET THE "DCV" TO THE ARRAY''           00008590        0936
  0770                                                                     00008591        0937
  0771      BINBCD(DOCUNO=DOCUNON)$                                        00008600        0938
  0771                                                                     00008601        0939
  0772      SEND(DOCUNON,1,6)$                                             00008610        0940
  0773      MESAGE(18)$''MAKE HEADER''                                     00008620        0941
  0774      COMPAIRS=18H(                  )$                              00008621        0942
  0775      BYTE($10$) (COMPAIRS)=1H(.)$''PLACE DECIMAL POINT''            00008630        0943
LINE T-O-O-O LONG

  0777      FOR D=1,1,DCVPCT($D$) $                                        00008650        0945
  0778       BEGIN                                                         00008660        0946
  0779      DDCCON=DCVCON($D$) $                                           00008662        0947
  0780      DDCWGT=DCVWGT($D$) $                                           00008664        0948
  0780        ''DO CONCEPT FIRST, AS IT'S THE EASIER ONE''                 00008670        0949
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0781        BINBCD(DCVCON($D$)=CONCONCEPTH) $                        00008690          0950
 0782        BYTE($2,4$)(CONPAIRS)=BYTE($2,4$)(CONCONCEPTH)$          00008700          0951
 0782        ''NOW DO THE WEIGHT (YECHCHCHCHCHCHCHCH)''               00008710          0952
 0783        CONWEIGHT=DCVWGT($D$)*1,0E3 $                            00008720          0953
 0784       BINBCD(CONWEIGHT=CONWEIGHTH)$                             00008730          0954
 0785        BYTE($9$)(CONPAIRS)=BYTE($2$)(CONWEIGHTH)$               00008740          0955
 0786        BYTE($11,3$)(CONPAIRS)=BYTE($3,3$)(CONWEIGHTH)$          00008750          0956
 0787        SEND(CONPAIRS,3,15)$                                     00008760          0957
 0788       END                                                       00008765          0958
 0789       CON3, MESAGE(33)$                                         00008770          0959
 0790       SWITCH CONA=(CON,OPT,OPT)$                                00008780          0960
 0791       ZILCH=YESNO(XXX)$                                         00008790          0961
 0792       GOTO CONA($ZILCH$) $                                      00008800          0962
 0792       '' E N D  OF "CON" FUNCTION''                             00008810          0963
 0792                                                                 00008820          0964
 0793       ENDF.''END-QUERY FUNCTION''                               00008830          0965
 0793       FOR A=ALL(QCV)$                                           00008840          0966
 0794        CORREL($A$)=0,0$                                         00008870          0967
 0795       FOR A=0,1,255$                                            00008890          0968
 0796        BEGIN                                                    00008900          0969
 0797         ACCNUM($A$)=0$                                          00008910          0970
 0798         TEMPID($A$)=0$                                          00008920          0971
 0799         RETRNK($A$)=0$                                          00008930          0972
 0800         BIBSUP($A$)=0$                                          00008940          0973
 0801         LASRET($A$)=0$                                          00008950          0974
 0802         CORCOE($A$)=0,0$                                        00008960          0975
 0803        END                                                      00008970          0976
 0804       NENT(TEMP)=0 $ ''RESET "NENT(TEMP)"''                     00008971          0977
 0805       FOR A=0,1,99$                                             00008980          0978
 0806        BEGIN                                                    00008990          0979
 0807         WRDENT($A$)=18H(                  )$                    00009000          0980
 0808         STMENT($A$)=12H(            )$                          00009010          0981
 0809         NONO($A$)=0$                                            00009020          0982
 0810        NNOT($A$)=18H(                  )$                       00009030          0983
 0811         QURENT($A$)=12H(            )$                          00009040          0984
 0812         DELWD($A$)=18H(                  )$                     00009041          0985
 0813        END                                                      00009050          0986
 0814       FOR A=1,1,4$''DON'T TOUCH "TERSE" PLEASE''                00009060          0987
 0815        MODEF($A$)=V(OFF)$                                       00009070          0988
 0816        NEWQ  =V(OFF)$                                           00009080          0989
 0817        WFLG  =V(OFF)$                                           00009090          0990
 0818        RFLG  =V(OFF)$                                           00009100          0991
 0819        SEEFLG=V(OFF)$                                           00009110          0992
 0820        DEFLG =V(OFF)$                                           00009120          0993
 0821        WRDFLG=V(OFF)$                                           00009130          0994
 0822       DOCDOC=V(OFF)$                                            00009140          0995
 0823       NULQRY=V(OFF)$                                            00009150          0996
 0824       OFFLG=V(UNUSED)$                                          00009160          0997
 0825       PREFIL=V(NO)$                                             00009170          0998
 0826       HITS  =0$                                                 00009180          0999
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0827       NEWTID=1$                                          00009181      1000
 0828       PLACE =6H(      )$                                 00009190      1001
 0829       NENT(.CVF)=0$                                      00009191      1002
 0830       NENT(QUERST)=0$                                    00009192      1003
 0831       NENT(NONOT)=0$                                     00009193      1004
 0832       NENT(NOTAB)=0$                                     00009194      1005
 0833       NENT(WORDS)=0$                                     00009195      1006
 0834       NENT(STEMS)=0$                                     00009196      1007
 0835       NENT(DELWDS)=0$                                    00009197      1008
 0836       NENT(COUNTS)=0$                                    00009198      1009
 0837       NENT(DELCNTS)=0$                                   00009199      1010
 0838       MESAGE(32)$                                        00009200      1011
 0839       GOTO DIA$$                                         00009210      1012
 0839                                                          00009220      1013
 0839       ''WGT FUNCTION''                                   00009230      1014
 0840       WGT. MESAGE(71)$                                   00009240      1015
 0841       WGT2. LLINK(6H(C/PARE))$                           00009250      1016
 0842       NENT(QCVF)=0$                                      00009251      1017
 0843       FOR Z=ALL(QCV)$                                    00009252      1018
 0844        BEGIN                                             00009253      1019
 0845         IF CORREL($Z$) EQ 0.0 $                          00009254      1020
 0846          TEST Z $                                        00009255      1021
 0847         QCVACH($NENT(QCVF)$)=Z$                          00009256      1022
 0848         QCVCON($NENT(QCVF)$)=CORREL($Z$) $               00009257      1023
 0849         NENT(QCVF)=NENT(QCVF)+1$                         00009258      1024
 0850        END                                               00009259      1025
 0851       IF NENT(QCVF) EQ 0 $                               00009260      1026
 0852        BEGIN                                             00009270      1027
 0853         SEND(6H(EMPTY ),1,5)$                            00009280      1028
 0854         GOTO WGT3$                                       00009290      1029
 0855        END                                               00009300      1030
 0856       MESAGE(18)$                                        00009310      1031
 0857       FOR B=ALL(QCVF)$                                   00009320      1032
 0858        BEGIN                                             00009330      1033
 0859         CWFIELD=12H(            )$                       00009340      1034
 0859                                                          00009341      1035
 0860       BINBCD(QCVACH($B$)=WGTCON)$                        00009350      1036
 0860                                                          00009351      1037
 0861         BYTE($0,4$)(CWFIELD)=BYTE($2,4$)(WGTCON) $       00009360      1038
 0862         FMTWGT(B,0) $                                    00009370      1039
 0863         SEND(CWFIELD,2,12)$                              00009380      1040
 0864        END                                               00009390      1041
 0865       IF SEEFLG EQ V(OFF)$                               00009510      1042
 0866        GOTO WGT3$                                        00009520      1043
 0867       SEEFLG=V(ON)$                                      00009530      1044
 0868       GOTO OPT$                                          00009540      1045
 0869       WGT3. MESAGE(61)$                                  00009550      1046
 0870       SWITCH WGTA=(,MOD12,OPT)$                          00009560      1047
 0871       ZILCH=YESNO(XXX)$                                  00009570      1048
 0872       GOTO WGTA($ZILCH$)$                                00009580      1049
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0873      MESAGE(62)$                                                        00009590          1050
  0874      WGT5, CWPAIR(=WGTCONCEP,WGTWEIGHT,INVALFLG)$                       00009600          1051
  0875      WGT3'1. IF INVALFLG EQ V(YES)$                                     00009610          1052
  0876       BEGIN                                                             00009620          1053
  0877        MESAGE(64)$                                                      00009630          1054
  0878        GOTO WGT4$                                                       00009640          1055
  0879       END                                                               00009650          1056
  0880       IF WGTCONCEP GR 1501 OR WGTCONCEP LQ 0 $                          00009660          1057
  0882       BEGIN INVALFLG = V(YES)$                                          00009670          1058
  0884       GOTO WGT3'1$ END                                                  00009680          1059
  0885      CORREL($WGTCONCEP$)=WGTWEIGHT$                                     00009800          1060
  0886      WGT4, MESAGE(3)$                                                   00009820          1061
  0887      SWITCH WGTB=(,MOD12,OPT)$                                          00009830          1062
  0888      ZILCH=YESNO(XXX)$                                                  00009840          1063
  0889      GOTO WGT3($ZILCH$)$                                                00009850          1064
  0890      MESAGE(63)$                                                        00009860          1065
  0891      GOTO WGT5 $                                                        00009870          1066
  0891      ''END OF WGT FUNCTION''                                            00009880          1067
  0892      PROC FMTWGT(BB,CC)$                                                00009890          1068
  0893      ITEM BB I 36 S $                                                   00009900          1069
  0894      ITEM CC I 36 S $                                                   00009910          1070
  0895      BEGIN                                                              00009920          1071
  0896       WGTCORF=QCVCOR($BB$) $                                            00009930          1072
  0897       WGTCOR=WGTCORF*1.0E3$''MOVE DECIMAL POINT''                       00009940          1073
  0898       IF WGTCOR LS 0$''NEGATIVE WEIGHT CHECK''                          00009950          1074
  0899        BEGIN                                                            00009960          1075
  0900        BYTE($CC+6$) (CWFIELD)=1H(-)$                                    00009970          1076
  0901        WGTCOR= -WGTCOR$                                                 00009980          1077
  0902        END                                                              00009990          1078
  0902                                                                         00009991          1079
  0903      BINBCD(WGTCOR=WGTCORH)$                                            00010000          1080
  0904      BYTE($CC+7$)(CWFIELD)=BYTE($2$)(WGTCORH)$                          00010010          1081
  0905       BYTE($CC+8$)(CWFIELD)=1H(.)$                                      00010020          1082
  0906      BYTE($9,3$)(CWFIELD)=BYTE($3,3$)(WGTCORH) $                        00010030          1083
  0906                                                                         00010031          1084
  0907       RETU*N$                                                           00010040          1085
  0908      END                                                                00010050          1086
  0909      JOBEND, ''ESCAPE FROM THE SYSTEMS (TERMINATE THE RUN)''            00010060          1087
  0909      IF OFFLG EQ V(USED)$                                               00010070          1088
  0910        GOTO JE2$''AN OFFLINE FILE WAS INDEED BUILT''                    00010080          1089
  0911      JE1, MESAGE(26)$                                                   00010090          1090
  0912            STOP $                                                       00010100          1091
  0913      JE2, SEND (30H(YOU HAVE AN OFF-LINE FILE
```

Figure VI. 28. (Continued)

```
0914          SEND(12H(PRESERVE IT),2,12)$                    00010120         1093
0915      SWITCH JEA=(,JE2'1,OPT)$                             00010130         1094
0916      ZILCH=YESNO(XXX)$                                    00010140         1095
0917      GOTO JEA($ZILCH$)$                                   00010150         1096
0918      SEND(6H(SPAWN),1,6)$                                00010160         1097
0919      SWITCH JEB=(JE7,JE4,OPT)$                            00010170         1098
0920      ZILCH=YESNO(XXX)$                                    00010171         1099
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08."67   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0921      GOTO JEB($ZILCH$) $                                        00010172      1100
  0922      JE2'1, SEND(12H(FILE PURGED ),2,12)$                       00010173      1101
  0923      GOTO JE1$                                                  00010174      1102
  0924      JE4, SEND(6H(PRINT),1,6) $                                00010175      1103
  0925      SWITCH JEC=(JE5,JE6,OPT)$                                  00010176      1104
  0926      ZILCH=YESNO(XXX)$                                          00010177      1105
  0927      GOTO JEC($ZILCH$) $                                        00010178      1106
  0928      JE5, COPYPR $''PRINT THE FILE''                            00010179      1107
  0929      SEND(12H(FILE PRINTED),2,12)$                              00010180      1108
  0930      GOTO JE1 $                                                 00010181      1109
  0931      JE6, COPYPP $''PUNCH THE FILE''                            00010182      1110
  0932      SEND(12H(FILE PUNCHED),2,12)$                              00010183      1111
  0933      GOTO JE1 $                                                 00010184      1112
  0934      JE7, SPAWN$ ''SPAWN A PRINT/PUNCH JOB''                    00010185      1113
  0935      GOTO JE1 $                                                 00010186      1114
  0935      ''END-OF "JOBEND" FUNCTION''                               00010187      1115
  0935      '' ********  "YESNO" -- VERSION FOUR ******** ''           00011380      1116
  0936      PROC YESNO(XXX)$ ''FUNCTION PROCEDURE''                    00011390      1117
  0937      ITEM REPLYA H  6 P 6H(      )$                             00011400      1118
  0938      ITEM REPLYB H  6 P 6H(      )$                             00011410      1119
  0939      ITEM YYES   H  6 P 6H(YE    )$                             00011420      1120
  0940      ITEM NNO    H  6 P 6H(NO    )$                             00011430      1121
  0941      ITEM OPTI   H  6 P 6H(OP    )$                             00011440      1122
  0942      ITEM WDS    I 36 S P 1 $                                   00011450      1123
  0943      ITEM CHS    I 36 S P 2 $                                   00011460      1124
  0944      ITEM YESNO I 36 S$''REPLY VALUE RETURNED HERE''            00011470      1125
  0945      BEGIN                                                      00011480      1126
  0946      YN1,                                                       00011490      1127
  0946      ''GET USER REPLY''                                         00011500      1128
  0946      RECV(=REPLYA,WDS,CHS)$                                     00011510      1129
  0946      ''NOW CHECK THE RESPONSE MADE''                            00011520      1130
  0947      BYTE($0,2$)(REPLYB)                                        00011530      1131
  0947       =                                                         00011540      1132
  0947      BYTE($0,2$)(REPLYA)$                                       00011550      1133
  0948      IFEITH                                                     00011560      1134
  0948       REPLYB EQ YYES$                                           00011570      1135
  0949        GOTO YES$                                                00011580      1136
  0950      ORIF                                                       00011590      1137
  0950       REPLYB EQ NNO$                                            00011600      1138
  0951        GOTO NO$                                                 00011610      1139
  0952      ORIF                                                       00011620      1140
  0952       REPLYB EQ OPTI$                                           00011630      1141
  0953        GOTO MAYBE$                                              00011640      1142
  0954      END                                                        00011650      1143
  0955      IF OPTION EQ V(ON)$                                        00011660      1144
  0956       BEGIN                                                     00011661      1145
  0957        PLACE=REPLYA$''MOVE USER'S REPLY TO "PLACE" IN "OPT" ''  00011662      1146
  0958        GOTO OPT7$                                               00011663      1147
  0959       END                                                       00011664      1148
  0959      '' ONLY "TYRO'S" EVER GET HERE ("OPTION" IS "OFF") ''      00011665      1149
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0960      MESAGE(1)$                                                         00011670         1150
  0961      GOTO YN1$''REPEAT SEQUENCE''                                       00011680         1151
  0963      YES. YESNO=0$ GOTO FIN$                                            00011690         1152
  0965      NO.  YESNO=1$ GOTO FIN$                                            00011700         1153
  0966      MAYBE. YESNO=2$                                                    00011710         1154
  0967      FIN. ZILCH=XXX$                                                    00011720         1155
  0968      END                                                                00011730         1156
  0968      '' "OFFLINE" FUNCTION CODE GOES HERE, BUT NOT IMPLEMENTED          00012960         1157
  0968          AS OF YET, OWING TO ITS LOW PRIORITY''                         00012970         1158
  0969      OFF. SEND(24H("OFF" NOT IMPLEMENTED   ),4,21) $                    00012980         1159
  0970           GOTO OPT$ ''RETURN TO OPTIONS ALLOCATOR''                     00012990         1160
  0971      BLD. SEND(30H(QUERY-CONCEPT-VECTOR BUILT....),5,30)$               00013000         1161
  0972      FOR Q=ALL(QCV)$                                                    00013010         1162
  0973       CORREL($Q$)=0.0 $                                                 00013020         1163
  0974      QCVBLD(QUERST=QCV)$                                                00013030         1164
  0975      GOTO OPT $                                                         00013040         1165
  0976      PROC LODCON $                                                      00013050         1166
  0977      BEGIN ''LODCON''                                                   00013060         1167
  0978      IN(1,CONTEN)$                                                      00013070         1168
  0979      NENT(CONTENTS)=0$                                                  00013080         1169
  0980      FOR C=0,1,499$                                                     00013090         1170
  0981       BEGIN                                                             00013100         1171
  0982        IO004(0,CONTEN,0,CONT($C$),2)$                                   00013110         1172
  0983        WAIT(IO004)$                                                     00013120         1173
  0984        IF CONTEN EQ V(EOF) $                                            00013130         1174
  0985         BEGIN                                                           00013140         1175
  0986          IN(4,CONTEN)$                                                  00013150         1176
  0987          NENT(CONTENTS)=C+1$                                            00013160         1177
  0988          RETURN$                                                        00013170         1178
  0989         END                                                             00013180         1179
  0990        END                                                              00013190         1180
  0991        NENT(CONTENTS)=500$''DEFAULT FOR TOO LARGE INPUT''               00013200         1181
  0992        RETURN$''SHOULD NEVER GET HERE, BUT JUST IN CASE''               00013210         1182
  0993      END ''LODCON''                                                     00013220         1183
  0994      PROC QCVBLD(QUERST=QCV)$                                           00013760         1184
  0995      TABLE QCV R 1502 S 1$                                              00013770         1185
  0995       BEGIN                                                             00013780         1186
  0996        ITEM CORREL F 0 0 $                                              00013790         1187
  0997       END                                                               00013800         1188
  0997      TABLE QUCWS V 4 S 1 $                                              00013810         1189
  0997       BEGIN                                                             00013820         1190
  0998        ITEM QUCW H 24 0 0 $                                             00013830         1191
  0999        ITEM QUCON I 18 S 0 0 $                                          00013840         1192
  1000        ITEM QUWGT A 18 S 14 0 18 $                                      00013850         1193
  1001       END                                                               00013860         1194
  1001      TABLE QUERST V 100 S 6 $                                           00013870         1195
  1001       BEGIN                                                             00013880         1196
  1002        ITEM QURENT H 12 0 0 $                                           00013890         1197
  1003        ITEM QUPAIRS H 24 2 0 $                                          00013900         1198
  1004       END                                                               00013910         1199
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NU.BER 46

  1004      BEGIN ''QCVBLD''                                                       00013920          1200
  1005      FOR A=ALL(QUERST)$                                                     00013930          1201
  1006       BEGIN                                                                 00013940          1202
  1007        QUCW($0$)=QUPAIRS($A$) $                                             00013950          1203
  1008        FOR B=0,1,3$''ADD TO GROWING "QCV"'                                  00013960          1204
  1009         CORREL($QUCON($B$)$)=CORREL($QUCON($B$)$)+QUWGT($B$)$               00014010          1205
  1010         END                                                                 00014020          1206
  1011        RETURN$                                                              00014030          1207
  1012      END                                                                    00014040          1208
  1012                                                                             00014370          1209
  1013      PROC CONTMP(=TEMP)$                                                    00014380          1210
  1013      ''COMPRESS "TEMP" TABLE ..                                             00014390          1211
  1013        REMOVES EMPTY SLOTS, AND RETURNS                                     00014400          1212
  1013        THE "NENT(TEMP)" REDUCED BY THE #                                    00014410          1213
  1013        OF COMPRESSED ENTRIES ''                                             00014420          1214
  1014      TABLE NULLS V 100 S 1 $                                                00014430          1215
  1014       BEGIN                                                                 00014440          1216
  1015        ITEM NULLA I 36 S 0 0 $                                              00014450          1217
  1016       END                                                                   00014460          1218
  1016                                                                             00014470          1219
  1016      TABLE TEMP V 256 S 2 $                                                 00014480          1220
  1016       BEGIN                                                                 00014490          1221
  1017        ITEM ACCNUM I 16 U 0  0 $                                            00014500          1222
  1018        ITEM TEMPID I  9 U 0 16 $                                            00014510          1223
  1019        ITEM RETRNK I  7 U 0 25 $                                            00014520          1224
  1020        ITEM BIBSUP I  1 U 0 32 $                                            00014530          1225
  1021        ITEM LASRET I  1 U 0 33 $                                            00014540          1226
  1022        ITEM CORCOE F      1  0 $                                            00014550          1227
  1023       END                                                                   00014560          1228
  1023      ITEM NULLHEAD I 36 S $                                                 00014570          1229
  1023                                                                             00014580          1230
  1024      BEGIN ''CONTMP''                                                       00014590          1231
  1025      NULLHEAD=0$                                                            00015000          1232
  1026      NENT(NULLS)=0$                                                         00015010          1233
  1027      FOR W=ALL(TEMP)$''PROCESS ALL TEMP ENTRIES''                           00015020          1234
  1028      BEGIN                                                                  00015030          1235
  1029       IF ACCNUM($W$) EQ 0 $                                                 00015040          1236
  1030        BEGIN                                                                00015060          1237
  1031         NULLA($NULLHEAD+NENT(NULLS)$)=W$''NULL POINTER CAPTURED''           00015070          1238
  1032         NENT(NULLS)=NENT(NULLS)+1$                                          00015080          1239
  1033         TEST W $                                                            00015090          1240
  1034        END                                                                  00015100          1241
  1035        IF NENT(NULLS) EQ 0 $                                                00015110          1242
  1036         TEST W $                                                            00015120          1243
  1036      ''ELSE, MOVE LAST GOOD ENTRY TO HEAD-OF-"NULLS" POSITION''             00015130          1244
  1037       ENTRY(TEMP($NULLA($NULLHEAD$)$))=ENTRY(TEMP($W$)) $                   00015140          1245
  1038       ENTRY(TEMP($W$))=0 $                                                  00015150          1246
  1039       NULLA($NULLHEAD+NENT(NULLS)$)=W $''NULL POINTER CAPTURED''            00015151          1247
  1040       NULLHEAD=NULLHEAD+1$                                                  00015160          1248
  1041      END ''"W" LOOP''                                                       00015180          1249
```

Figure VI.28. (Continued)

```
0023T 01  05-14-71   08,567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

   1042     FOR W=NENT(TEMP),-1,0$                                              00015190                      1250
   1043      BEGIN                                                              00015192                      1251
   1044       IF ACCNUM($W$) EQ 0 $                                             00015193                      1252
   1045        BEGIN                                                            00015194                      1253
   1046         NENT(TEMP)=W+1 $                                                00015195                      1254
   1047         RETURN$                                                         00015196                      1255
   1048        END                                                              00015197                      1256
   1049      END                                                                00015198                      1257
   1050     RETURN$                                                             00015200                      1258
   1051     END ''CONTMP''                                                      00015210                      1259
   1052     PROC COPYPR$                                                        00015430                      1260
   1053     ITEM MSG H 18 P 18H("OFFLINE" PRINT-SM)$                            00015440                      1261
   1054     BEGIN                                                               00015450                      1262
   1055      SEND(MSG,3,18)$                                                    00015460                      1263
   1056      RETURN$                                                            00015470                      1264
   1057     END                                                                 00015480                      1265
   1057                                                                         00015490                      1266
   1058     PROC COPYPP$                                                        00015500                      1267
   1059     ITEM MSG H 18 P 18H("OFFLINE" PUNCH-SM)$                            00015510                      1268
   1060     BEGIN                                                               00015520                      1269
   1061      SEND(MSG,3,18)$                                                    00015530                      1270
   1062      RETURN$                                                            00015540                      1271
   1063     END                                                                 00015550                      1272
   1063                                                                         00015560                      1273
   1064     PROC SPAWN $                                                        00015570                      1274
   1065     ITEM MSG H 18 P 18H(NEWJOB "SPAWNED"  )$                            00015580                      1275
   1066     BEGIN                                                               00015590                      1276
   1067      SEND(MSG,3,18)$                                                    00015600                      1277
   1068      RETURN$                                                            00015610                      1278
   1069     END                                                                 00015620                      1279
   1069                                                                         00015820                      1280
   1070     PROC  SNDABS(ABNO)$                                                 00020000                      1281
   1070                                                                         00020010                      1282
   1071     ITEM ABNO I 36 S $                                                  00020020                      1283
   1072     ITEM LINENDF I 36 S P O(770120202020)$                              00020030                      1284
   1073     ITEM LINEND H 6 $                                                   00020040                      1285
   1074     OVERLAY LINEND=LINENDF $                                            00020050                      1286
   1075     ITEM NUMB I 36 S $                                                  00020060                      1287
   1076     ARRAY ABSTRACT 11 30 H 6 $                                          00020070                      1288
   1077     OVERLAY NUMB,ABSTRACT $                                             00020080                      1289
   1078     BEGIN ''SNDABS.. SEND ABSTRACT TO TERMINAL USER ''                  00020090                      1290
   1079     RETDOC(ABNO=NUMB)$''GET THE ABSTRACT FROM RANDOM DISC''             00020100                      1291
   1080     FOR C=0,1,29$                                                       00020110                      1292
   1081      BEGIN ''C''                                                        00020120                      1293
   1082       FOR R=0,1,10$                                                     00020130                      1294
   1083        BEGIN ''R''                                                      00020140                      1295
   1084         IF ABSTRACT($R,C$) EQ 6H(000000)$                               00020150                      1296
   1085          BEGIN ''INNER LOOP''                                           00020160                      1297
   1086           ABSTRACT($R,C$) = LINEND$                                     00020170                      1298
   1087           SEND(ABSTRACT($0,C$),11,66)$''OUT IT GOES''                   00020180                      1299
```

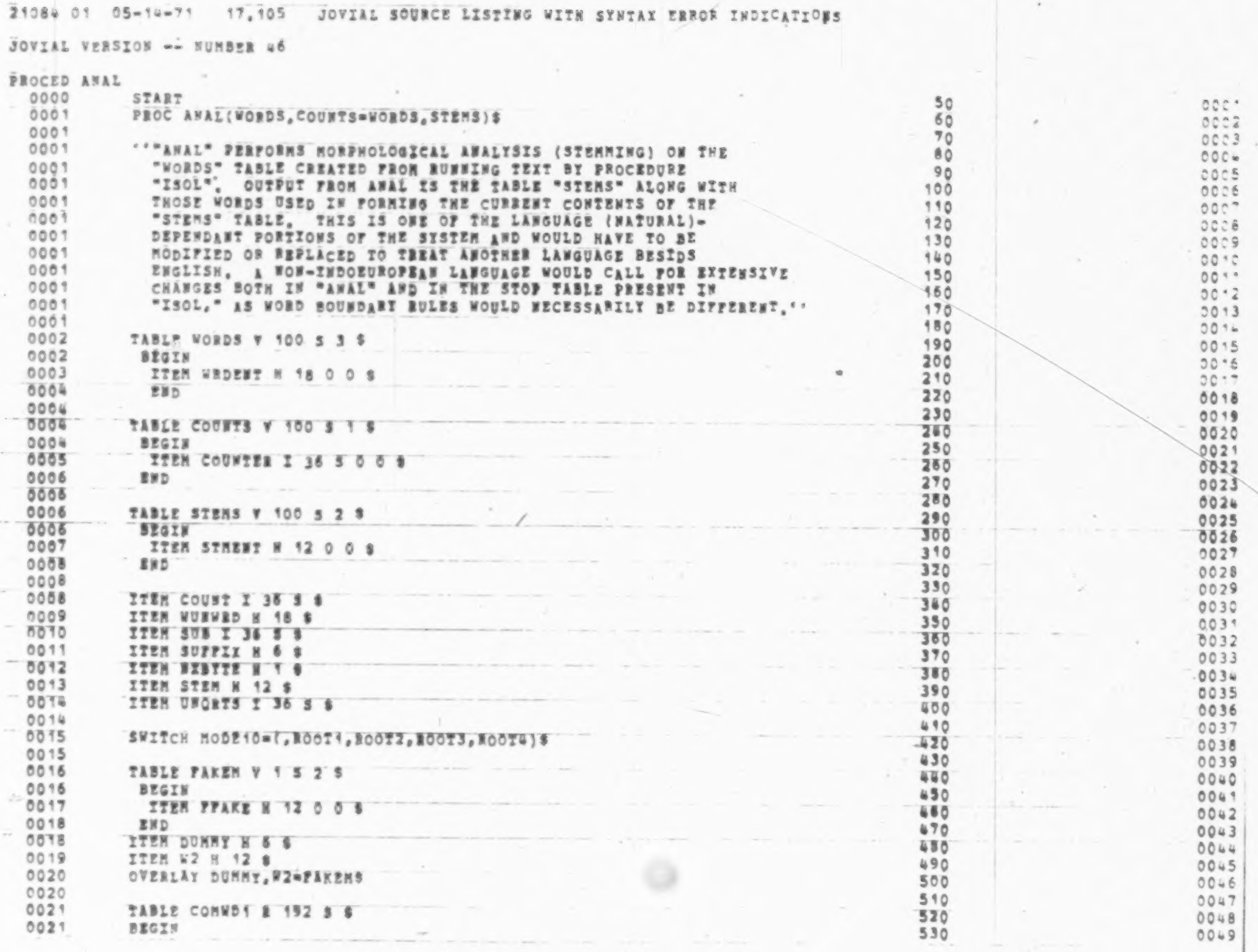
Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  1088               RETURN $                                                      00020190          1300
  1089             END ''INNER LOOP''                                              00020200          1301
  1090           END ''R''                                                         00020210          1302
  1090          ''NO ZERO WORDS .. SEND THE WHOLE LINE THIS ITERATION''            00020220          1303
  1091          SEND(ABSTRACT($0,C$),11,66)$                                       00020230          1304
  1092         END ''C''                                                           00020240          1305
  1093        RETURNS''WE SHOULD "NEVER" GET HERE, BUT JUST IN CASE ''             00020250          1306
  1094        END ''SNDABS''                                                       00020260          1307
  1094                                                                             00020270          1308
  1094                                                                             00020280          1309
  1094                                                                             00020290          1310
  1095        PROC  SNDBIB(BINO)$                                                  00020300          1311
  1095                                                                             00020310          1312
  1096        ITEM BINO I 36 S $                                                   00020320          1313
  1097        ITEM LINENDF I 36 S P O(770120202020) $                              00020330          1314
  1098        ITEM LINEND H 6 $                                                    00020340          1315
  1099        OVERLAY LINEND=LINENDF $                                             00020350          1316
  1100        ITEM NUMB I 36 S $                                                   00020360          1317
  1101        ARRAY BIBLIO 11 30 H 6 $                                             00020370          1318
  1102        OVERLAY NUMB,BIBLIO $                                                00020380          1319
  1103        BEGIN ''SNDBIB .. SEND BIBLIOGRAPHY TO TERMINAL USER ''              00020390          1320
  1104        FETBIB(BINO=NUMB)$''GET THE BIBLIOGRAPHY FROM DISC''                 00020400          1321
  1105        FOR C=0,1,29$                                                        00020410          1322
  1106         BEGIN ''C''                                                         00020420          1323
  1107          FOR R=0,1,10$                                                      00020430          1324
  1108           BEGIN ''R''                                                       00020440          1325
  1109            IF BIBLIO($R,C$) EQ 6H(000000)$                                  00020450          1326
  1110             BEGIN ''INNER LOOP''                                            00020460          1327
  1111              BIBLIO($R,C$)=LINEND$                                          00020470          1328
  1112              SEND(BIBLIO($0,C$),11,66)$''AWAY IT GOES''                     00020480          1329
  1113              RETURN $                                                       00020490          1330
  1114             END ''INNER LOOP''                                              00020500          1331
  1115           END ''R''                                                         00020510          1332
  1115          ''NO ZERO WORDS ..., SEND THE WHOLE LINE THIS ITERATION''          00020520          1333
  1116          SEND(BIBLIO($0,C$),11,66)$                                         00020530          1334
  1117         END ''C''                                                           00020540          1335
  1118        RETURNS''WE SHOULD "NEVER" GET HERE, BUT JUST IN CASE ''             00020550          1336
  1119        END ''SNDBIB''                                                       00020560          1337
  1119                                                                             00020570          1338
  1119                                                                             00020580          1339
  1120        SKIM. LLINK(6H(CWPARE))$                                             00021000          1340
  1121         CWPAIR(=WGTCONCEP,WGTWEIGHT,INVALFLG) $                             00021010          1341
  1122         IF INVALFLG EQ V(YES) $                                             00021020          1342
  1123          GOTO OPTS ''INVALID INPUT FROM "CWPAIR" !'                         00021030          1343
  1124         FOR S=ALL(TEMP) $ ''REMOVE "SKIMMED" "TEMP" ENTRIES''               00021040          1344
  1125          BEGIN                                                              00021050          1345
  1126           IF CORCOE($S$) LS WGTWEIGHT $                                     00021060          1346
  1127            BEGIN                                                            00021070          1347
  1128             ENTRY(TEMP($S$))=0$                                             00021080          1348
  1129             CORCOE($S$)=0.0 $                                               00021090          1349
```

Figure VI. 28. (Continued)

```
0023T 01  05-14-71   08.567   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 1130          END
 1131         END                                                             00021100              1350
 1132        COMTMP(=TEMP) $''COMPRESS THE "TEMP" TABLE''                     00021110              1351
 1133        BINBCD(NENT(TEMP)=TEMPH) $                                       00021120              1352
 1134        BYTE($0,6$)(SKIMESS)=TEMPH$                                      00021130              1353
 1135        SEND(SKIMESS,4,24)$                                              00021140              1354
 1136        GOTO OPTS ''AND RETURN TO ALLOCATOR''                            00021150              1355
 1137       TERM $                                                            00021160              1356
                                                                              00050000              1357
```

Figure VI. 28. (Concluded)



```
21084 01  05-14-71   17,105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROCED ANAL
  0000      START                                                                  50      0001
  0001      PROC ANAL(WORDS,COUNTS=WORDS,STEMS)$                                   60      0002
  0001                                                                             70      0003
  0001      ''"ANAL" PERFORMS MORPHOLOGICAL ANALYSIS (STEMMING) ON THE             80      0004
  0001        "WORDS" TABLE CREATED FROM RUNNING TEXT BY PROCEDURE                 90      0005
  0001        "ISOL". OUTPUT FROM ANAL IS THE TABLE "STEMS" ALONG WITH            100      0006
  0001        THOSE WORDS USED IN FORMING THE CURRENT CONTENTS OF THE             110      0007
  0001        "STEMS" TABLE.  THIS IS ONE OF THE LANGUAGE (NATURAL)-              120      0008
  0001        DEPENDANT PORTIONS OF THE SYSTEM AND WOULD HAVE TO BE               130      0009
  0001        MODIFIED OR REPLACED TO TREAT ANOTHER LANGUAGE BESIDS               140      0010
  0001        ENGLISH.  A NON-INDOEUROPEAN LANGUAGE WOULD CALL FOR EXTENSIVE      150      0011
  0001        CHANGES BOTH IN "ANAL" AND IN THE STOP TABLE PRESENT IN             160      0012
  0001        "ISOL," AS WORD BOUNDARY RULES WOULD NECESSARILY BE DIFFERENT.''    170      0013
  0001                                                                            180      0014
  0002      TABLE WORDS V 100 S 3 $                                               190      0015
  0002       BEGIN                                                                200      0016
  0003        ITEM WRDENT H 18 0 0 $                                              210      0017
  0004       END                                                                  220      0018
  0004                                                                            230      0019
  0004      TABLE COUNTS V 100 S 1 $                                              240      0020
  0004       BEGIN                                                                250      0021
  0005        ITEM COUNTER I 36 S 0 0 $                                           260      0022
  0006       END                                                                  270      0023
  0006                                                                            280      0024
  0006      TABLE STEMS V 100 S 2 $                                               290      0025
  0006       BEGIN                                                                300      0026
  0007        ITEM STMENT H 12 0 0 $                                              310      0027
  0008       END                                                                  320      0028
  0008                                                                            330      0029
  0008      ITEM COUNT I 36 S $                                                   340      0030
  0009      ITEM WUNWRD H 18 $                                                    350      0031
  0010      ITEM SUB I 36 S $                                                     360      0032
  0011      ITEM SUFFIX H 6 $                                                     370      0033
  0012      ITEM NXBYTE H 1 $                                                     380      0034
  0013      ITEM STEM H 12 $                                                      390      0035
  0014      ITEM UNQRTS I 36 S $                                                  400      0036
  0014                                                                            410      0037
  0015      SWITCH MODE10=(,ROOT1,ROOT2,ROOT3,ROOT4)$                             420      0038
  0015                                                                            430      0039
  0016      TABLE FAKEM V 1 S 2 $                                                 440      0040
  0016       BEGIN                                                                450      0041
  0017        ITEM FFAKE H 12 0 0 $                                               460      0042
  0018       END                                                                  470      0043
  0018      ITEM DUMMY H 6 $                                                      480      0044
  0019      ITEM W2 H 12 $                                                        490      0045
  0020      OVERLAY DUMMY,W2=FAKEM$                                               500      0046
  0020                                                                            510      0047
  0021      TABLE COMWD1 R 192 S $                                                520      0048
  0021      BEGIN                                                                 530      0049
```

Figure VI.29. ANALyze Listing

```
21084 01  05-10-71   17,105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0022      ITEM COM1 H 6 $                                                    540         0050
  0022      BEGIN                                                              550         0051
  0022        6H(A     ) 6H(ABOUT ) 6H(ABOVE )                                 560         0052
  0022        6H(ACROSS) 6H(AFTER ) 6H(ALL   )                                 570         0053
  0022        6H(ALMOST) 6H(ALONE ) 6H(ALONG )                                 580         0054
  0022        6H(ALSO  ) 6H(ALWAYS) 6H(AM    )                                 590         0055
  0022        6H(AMONG ) 6H(AN    ) 6H(AND   )                                 600         0056
  0022        6H(ANY   ) 6H(ANYONE) 6H(APART )                                 610         0057
  0022        6H(ARE   ) 6H(AROUND) 6H(AS    )                                 620         0058
  0022        6H(ASIDE ) 6H(AT    ) 6H(AWAY  )                                 630         0059
  0022        6H(BE    ) 6H(BEEN  ) 6H(BEFORE)                                 640         0060
  0022        6H(BEHIND) 6H(BEGIN ) 6H(BELOW )                                 650         0061
  0022        6H(BEYOND) 6H(BOTH  ) 6H(BUT   )                                 660         0062
  0022        6H(BY    ) 6H(CAN   ) 6H(CANNOT)                                 670         0063
  0022        6H(COULD ) 6H(DID   ) 6H(DO    )                                 680         0064
  0022        6H(DOES  ) 6H(DOING ) 6H(DONE  )                                 690         0065
  0022        6H(DOWN  ) 6H(DURING) 6H(EACH  )                                 700         0066
  0022        6H(EITHER) 6H(ELSE  ) 6H(ENOUGH)                                 710         0067
  0022        6H(ETC   ) 6H(EVEN  ) 6H(EVER  )                                 720         0068
  0022        6H(EVERY ) 6H(EXCEPT) 6H(FEW   )                                 730         0069
  0022        6H(FOR   ) 6H(FORTH ) 6H(FROM  )                                 740         0070
  0022        6H(GET   ) 6H(GETS  ) 6H(GOT   )                                 750         0071
  0022        6H(HAD   ) 6H(HARDLY) 6H(HAS   )                                 760         0072
  0022        6H(HAVE  ) 6H(HAVING) 6H(HENCE )                                 770         0073
  0022        6H(HE    ) 6H(HER   ) 6H(HERE  )                                 780         0074
  0022        6H(HEREIN) 6H(HERS  ) 6H(HIM   )                                 790         0075
  0022        6H(HIS   ) 6H(HITHER) 6H(HOW   )                                 800         0076
  0022        6H(INDEED) 6H(INTER ) 6H(INTO  )                                 810         0077
  0022        6H(INWARD) 6H(IS    ) 6H(IT    )                                 820         0078
  0022        6H(ITSELF) 6H(ITS   ) 6H(IT'S  )                                 830         0079
  0022        6H(JUST  ) 6H(KEEP  ) 6H(KEPT  )                                 840         0080
  0022        6H(LEAST ) 6H(LESS  ) 6H(LEST  )                                 850         0081
  0022        6H(MANY  ) 6H(MAY   ) 6H(ME    )                                 860         0082
  0022        6H(MIGHT ) 6H(MINE  ) 6H(MORE  )                                 870         0083
  0022        6H(MOST  ) 6H(MUCH  ) 6H(MUST  )                                 880         0084
  0022        6H(MY    ) 6H(MYSELF) 6H(NEXT  )                                 890         0085
  0022        6H(NO    ) 6H(NOBODY) 6H(NONE  )                                 900         0086
  0022        6H(NOR   ) 6H(NOT   ) 6H(OF    )                                 910         0087
  0022        6H(OH    ) 6H(ON    ) 6H(ONE   )                                 920         0088
  0022        6H(ONES  ) 6H(ONLY  ) 6H(ONTO  )                                 930         0089
  0022        6H(OR    ) 6H(OTHER ) 6H(OTHERS)                                 940         0090
  0022        6H(OUGHT ) 6H(OUR   ) 6H(OURS  )                                 950         0091
  0022        6H(OVER  ) 6H(OWN   ) 6H(PER   )                                 960         0092
  0022        6H(PLEASE) 6H(PLUS  ) 6H(QUITE )                                 970         0093
  0022        6H(RATHER) 6H(REALLY) 6H(RIGHT )                                 980         0094
  0022        6H(SELF  ) 6H(SELVES) 6H(SHALL )                                 990         0095
  0022        6H(SHE   ) 6H(SHOULD) 6H(SINCE )                                1000         0096
  0022        6H(SIX   ) 6H(SO    ) 6H(SOME  )                                1010         0097
  0022        6H(STILL ) 6H(SUCH  ) 6H(TEN   )                                1020         0098
  0022        6H(THAN  ) 6H(THAT  ) 6H(THE   )                                1030         0099
```

Figure VI. 29. (Continued)

```
 21084 01  05-14-71   17,105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0022       6H(THEIR ) 6H(THEIRS) 6H(THEM   )                                         1040          0100
  0022       6H(THEN  ) 6H(THENCE) 6H(THERE  )                                         1050          0101
  0022       6H(THESE ) 6H(THEY  ) 6H(THIS   )                                         1060          0102
  0022       6H(THOSE ) 6H(THOUGH) 6H(THUS   )                                         1070          0103
  0022       6H(TO    ) 6H(TOO   ) 6H(TOWARD)                                          1080          0104
  0022       6H(TWO   ) 6H(UNDER ) 6H(UNLESS)                                          1090          0105
  0022       6H(UNTIL ) 6H(UNTO  ) 6H(UP     )                                         1100          0106
  0022       6H(UPON  ) 6H(UPWARD) 6H(US     )                                         1110          0107
  0022       6H(VERY  ) 6H(WAS   ) 6H(WELL   )                                         1120          0108
  0022       6H(WE    ) 6H(WERE  ) 6H(WHAT   )                                         1130          0109
  0022         6H(WHEN  ) 6H(WHENCE) 6H{WHERE )                                        1140          0110
  0022       6H(WHICH ) 6H(WHILE ) 6H(WHO    )                                         1150          0111
  0022       6H(WHOM  ) 6H(WHOSE ) 6H(WHY    )                                         1160          0112
  0022       6H(WILL  ) 6H(WITH  ) 6H(WITHIN)                                          1170          0113
  0022       6H(WONT  ) 6H(WOULD ) 6H(YES    )                                         1180          0114
  0022       END                                                                       1190          0115
  0023      END                                                                        1200          0116
  0023                                                                                 1210          0117
  0023      TABLE COMWD2 R 47 S $                                                      1220          0118
  0023       BEGIN                                                                     1230          0119
  0024        ITEM COM2 H 12 $                                                         1240          0120
  0024         BEGIN                                                                   1250          0121
  0024        12H(AGAINST     ) 12H(ALTHOUGH    )                                      1260          0122
  0024        12H(ANOTHER     ) 12H(ANYBODY     )                                      1270          0123
  0024        12H(ANYTHING    ) 12H(ANYWHERE    )                                      1280          0124
  0024        12H(AWFULLY     ) 12H(BECAUSE     )                                      1290          0125
  0024        12H(BETWEEN     ) 12H(ELSEWHERE   )                                      1300          0126
  0024        12H(EVERYONE    ) 12H(EVERYTHING  )                                      1310          0127
  0024        12H(EVERYWHERE  ) 12H(FURTHERMORE )                                      1320          0128
  0024        12H(HERSELF     ) 12H(HIMSELF     )                                      1330          0129
  0024        12H(HOWBEIT     ) 12H(HOWEVER     )                                      1340          0130
  0024        12H(INASMUCH    ) 12H(INSOFAR     )                                      1350          0131
  0024        12H(INSTEAD     ) 12H(MOREOVER    )                                      1360          0132
  0024        12H(NEITHER     ) 12H(NEVERTHELESS)                                      1370          0133
  0024        12H(NOTHING     ) 12H(NOWHERE     )                                      1380          0134
  0024        12H(OTHERWISE   ) 12H(OURSELVES   )                                      1390          0135
  0024        12H(SOMEBODY    ) 12H(SOMETHING   )                                      1400          0136
  0024        12H(SOMETIMES   ) 12H(SOMEWHAT    )                                      1410          0137
  0024        12H(THEMSELVES  ) 12H(THEREBY     )                                      1420          0138
  0024        12H(THEREFORE   ) 12H(THROUGHOUT  )                                      1430          0139
  0024        12H(TOGETHER    ) 12H(UNDERNEATH  )                                      1440          0140
  0024        12H(WHATEVER    ) 12H(WHENEVER    )                                      1450          0141
  0024        12H(WHEREVER    ) 12H(WHETHER     )                                      1460          0142
  0024        12H(WITHOUT     ) 12H(YOURSELF    )                                      1470          0143
  0024        12H(YOURSELVES  )                                                        1480          0144
  0024         END                                                                     1490          0145
  0025      END                                                                        1500          0146
  0025                                                                                 1510          0147
  0025      TABLE SUP1TB R 5 S $                                                       1520          0148
  0025       BEGIN                                                                     1530          0149
```

Figure VI. 29. (Continued)

```
21084 01  05-14-71   17.105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0026        ITEM SUF1 H 6 $                                                        1540        0150
  0026         BEGIN                                                                 1550        0151
  0026         6H(E00000) 6H(I00000) 6H(S0/000) 6H('00000) 6H(Y00000)                1560        0152
  0026         END                                                                   1570        0153
  0027        END                                                                    1580        0154
  0027                                                                               1590        0155
  0027       TABLE SUF2TB R 16 S $                                                   1600        0156
  0027        BEGIN                                                                  1610        0157
  0028         ITEM SUF2 H 6 $                                                       1620        0158
  0028          BEGIN                                                                1630        0159
  0028          6H(AL0000) 6H(AN0000) 6H(AR0000) 6H(CY0000) 6H(ED0000)               1640        0160
  0028          6H(EN0000) 6H(ER0000) 6H(ET0000) 6H(IC0000) 6H(LY0000)               1650        0161
  0028          6H(ON0000) 6H(OR0000) 6H(OU0000) 6H(RY0000) 6H('S0000)               1660        0162
  0028          6H(TH0000)                                                           1670        0163
  0028          END                                                                  1680        0164
  0029        END                                                                    1690        0165
  0029                                                                               1700        0166
  0029       TABLE SUF3TB R 44 S $                                                   1710        0167
  0029        BEGIN                                                                  1720        0168
  0030         ITEM SUF3 H 6 $                                                       1730        0169
  0030         BEGIN                                                                 1740        0170
  0030        6H(AGE000) 6H(ANT000) 6H(ART000) 6H(ATE000) 6H(BAR000)                 1750        0171
  0030        6H(CAN000) 6H(DER000) 6H(END000) 6H(ENT000) 6H(EST000)                 1760        0172
  0030        6H(ETH000) 6H(FUL000) 6H(GEN000) 6H(IAL000) 6H(IAN000)                 1770        0173
  0030        6H(IED000) 6H(IES000) 6H(ING000) 6H(ISH000) 6H(ISM000)                 1780        0174
  0030        6H(ION000) 6H(IST000) 6H(ITY000) 6H(IVE000) 6H(IZE000)                 1790        0175
  0030        6H(LAY000) 6H(LAY000) 6H(LEL000) 6H(MAN000) 6H(MEN000)                 1800        0176
  0030        6H(NOT000) 6H(ODE000) 6H(OSE000) 6H(OUS000) 6H(PLY000)                 1810        0177
  0030        6H(STY000) 6H(TAL000) 6H(TER000) 6H(TIC000) 6H(TLE000)                 1820        0178
  0030        6H(ULE000) 6H(URE000) 6H(VAR000) 6H(WAY000)                            1830        0179
  0030          END                                                                  1840        0180
  0031        END                                                                    1850        0181
  0031                                                                               1860        0182
  0031       TABLE SUF4TB R 30 S $                                                   1870        0183
  0031        BEGIN                                                                  1880        0184
  0032         ITEM SUF4 H 6 $                                                       1890        0185
  0032          BEGIN                                                                1900        0186
  0032        6H(ABLE00) 6H(ANCE00) 6H(CANT00) 6H(CIDE00) 6H(DUCE00)                 1910        0187
  0032        6H(ENCE00) 6H(EVER00) 6H(HAND00) 6H(IENT00) 6H(ITIE00)                 1920        0188
  0032        6H(LENT00) 6H(LESS00) 6H(MATE00) 6H(MENT00) 6H(MITY00)                 1930        0189
  0032        6H(NESS00) 6H(PEAR00) 6H(SERT00) 6H(SHIP00) 6H(SING00)                 1940        0190
  0032        6H(SONE00) 6H(TEEN00) 6H(THER00) 6H(TIAL00) 6H(TION00)                 1950        0191
  0032        6H(TIZE00) 6H(TURE00) 6H(VENT00) 6H(VICE00) 6H(WISE00)                 1960        0192
  0032          END                                                                  1970        0193
  0033        END                                                                    1980        0194
  0033                                                                               1990        0195
  0033       TABLE ELIM R 64 S $                                                     2000        0196
  0033        BEGIN.                                                                 2010        0197
  0034         ITEM ELM I 36 S $                                                     2020        0198
  0034          BEGIN                                                                2030        0199
```

Figure VI.29. (Continued)

```
21084 01  05-14-71    17,105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0034        0 0 0 0 0 0 0 0                                                   2040        0200
  0034        0 0 0 0 0 0 0 0                                                   2050        0201
  0034        0 1 0 0 0 1 0 0                                                   2060        0202
  0034        0 1 0 0 0 0 0 0                                                   2070        0203
  0034        0 0 0 0 0 0 1 0                                                   2080        0204
  0034        0 0 0 0 0 0 0 0                                                   2090        0205
  0034        0 0 0 0 1 0 0 0                                                   2100        0206
  0034        0 0 0 0 0 0 0 0                                                   2110        0207
  0034          END                                                             2120        0208
  0035        END                                                               2130        0209
  0035                                                                          2140        0210
  0035       BEGIN ''MORPHOLOGICAL ANALYSIS (STEM REMOVAL)''                    2150        0211
  0035       ''INITIALIZE''                                                     2160        0212
  0036       UNORTS=0$                                                          2170        0213
  0037       COUNT=0$                                                           2180        0214
  0038       NXBYTE=1H( )$                                                      2200        0215
  0039       FIND=0$                                                            2220        0216
  0040       DIRECT                                                             2230        0217
  0041              LDX0    5,1                                                 2240        0218
  0042              ADX0    1,DU                                                2250        0219
  0043              STX0    FSTEM                                               2260        0220
  0044       JOVIAL                                                             2270        0221
  0045       SUFFIX=6H(000000)$                                                 2271        0222
  0046       STEM=12H(000000000000)$                                            2272        0223
  0046       ''ENTER MAIN LOOP ... PROCESS                                      2280        0224
  0046         "WORDS" INTO "STEMS" ''                                          2290        0225
  0047       FOR W=ALL(WORDS)$                                                  2300        0226
  0048       BEGIN ''"W"= LOOP''                                                2310        0227
  0048                                                                          2320        0228
  0049       COUNT=COUNTER($W$) $                                               2330        0229
  0050       WUNWRD=WRDENT($W$) $                                               2340        0230
  0051       IFEITH                                                             2350        0231
  0051         COUNT GR 12 $                                                    2360        0232
  0052          GOTO LOOP12$''CANNOT BE "COMMON" > 12 CHARACTERS''              2370        0233
  0053        ORIF                                                              2380        0234
  0053         COUNT GR 6 $''BETWEEN 6 & 12 CHARACTERS''                        2390        0235
  0054          GOTO PLEBE2$                                                    2400        0236
  0055        ORIF                                                              2410        0237
  0055         COUNT LQ 1 $                                                     2420        0238
  0056          TEST W $''SINGLETON GARBAGE LETTERS AND "A" ''                  2430        0239
  0057       END                                                                2440        0240
  0057       ''ELSE THIS IS "PLEBE1" ,, TRY COMWD1''                            2450        0241
  0058       HUNT1(WUNWRD,COMWD1,0,NENT(COMWD1)-1=FIND)$                        2460        0242
  0059        IF FIND GQ 0 $''FOUND IT, IT'S COMMON''                           2470        0243
  0060         TEST W $                                                         2480        0244
  0061        IF COUNT LQ 4 $                                                   2490        0245
  0062         GOTO ROOT8$                                                      2500        0246
  0063         GOTO LOOP12$                                                     2510        0247
  0064       PLEBE2.''ANALYSIS OF WORDS > 12 CHARACTERS IN LENGTH''             2520        0248
  0064        HUNT2(WUNWRD,COMWD2,0,NENT(COMWD2)-1=FIND)$                       2530        0249
```

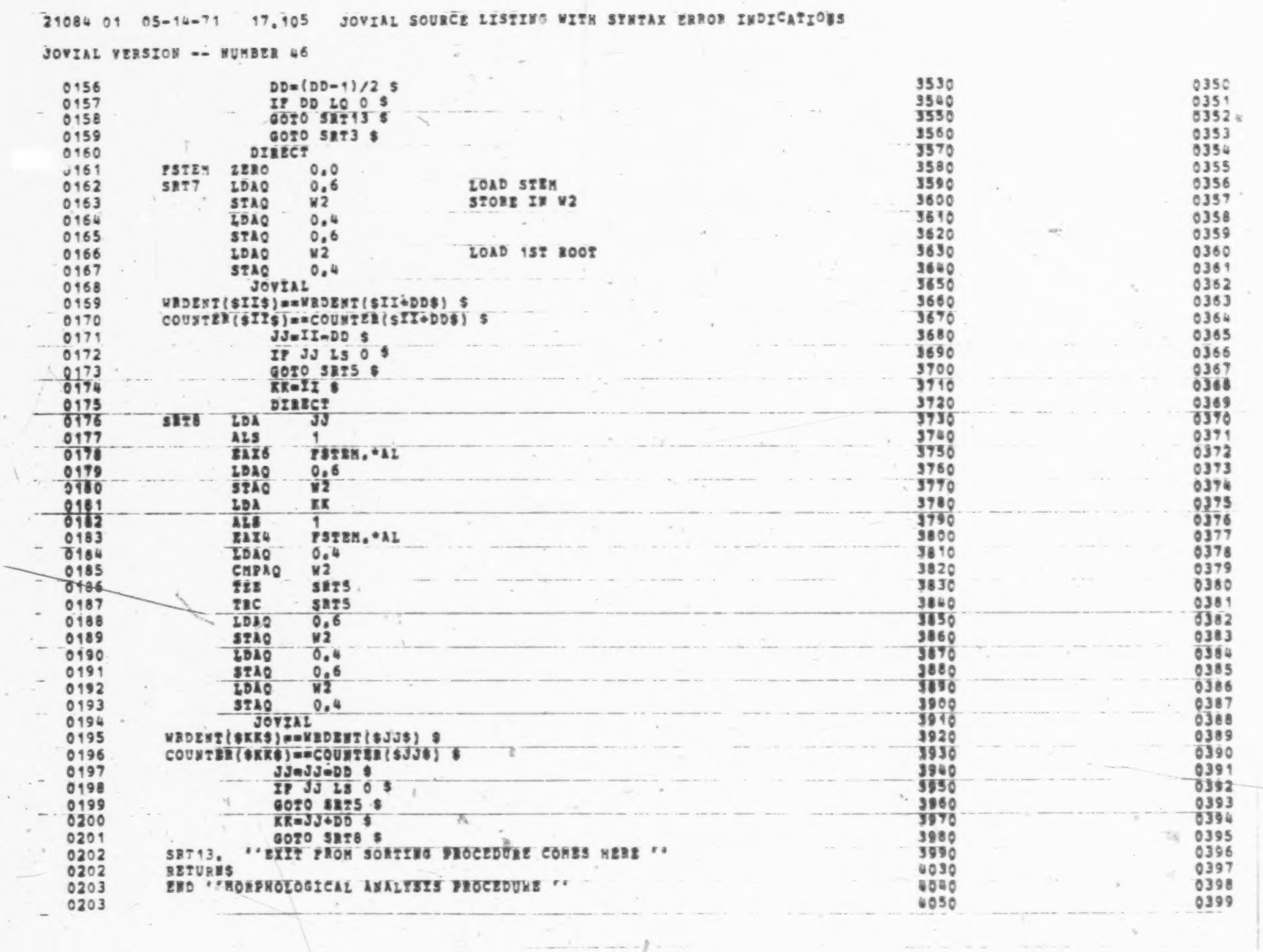
Figure VI.29. (Continued)

```
21084 01  05-14-71   17,105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0065         IF FIND GQ 0 $''FOUND IT, IT'S COMMON''                     2540      0250
 0066          TEST W $                                                   2550      0251
 0067       LOOP12. SUB=5 $                                               2560      0252
 0068       LOOP13. SUB=SUB-1$                                            2570      0253
 0069       DIRECT                                                        2580      0254
 0070              STZ      SUFFIX                                        2590      0255
 0071       JOVIAL                                                        2600      0256
 0072       BYTE($0,SUB$)(SUFFIX)=BYTE($COUNT-SUB,SUB$)(WUNWRD)$          2610      0257
 0073       GOTO MODE10($SUB$) $                                          2620      0258
 0074        TEST W $''DEFAULT "BOMB-OUT" ''                              2630      0259
 0075       ROOT1.''STEM CANDIDATES, LENGTH 1''                           2640      0260
 0075        HUNT1(SUFFIX,SUF1TB,0,NENT(SUF1TB)-1=FIND)$                  2650      0261
 0076        IF FIND GQ 0 $                                               2660      0262
 0077         GOTO ROOT6 $                                                2670      0263
 0078         GOTO ROOT8 $                                                2680      0264
 0079       ROOT2.''STEM CANDIDATES, LENGTH 2''                           2690      0265
 0079        HUNT1(SUFFIX,SUF2TB,0,NENT(SUF2TB)-1=FIND)$                  2700      0266
 0080         IF FIND GQ 0 $                                              2710      0267
 0081          GOTO ROOT6$                                                2720      0268
 0082          GOTO LOOP13$                                               2730      0269
 0083       ROOT3.''STEM CANDIDATES, LENGTH 3''                           2740      0270
 0083        HUNT1(SUFFIX,SUF3TB,0,NENT(SUF3TB)-1=FIND)$                  2750      0271
 0084         IF FIND GQ 0 $                                              2760      0272
 0085          GOTO ROOT6$                                                2770      0273
 0086          GOTO LOOP13$                                               2780      0274
 0087       ROOT4.''STEM CANDIDATES, LENGTH 4''                           2790      0275
 0087        HUNT1(SUFFIX,SUF4TB,0,NENT(SUF4TB)-1=FIND)$                  2800      0276
 0088         IF FIND LS 0 $                                              2810      0277
 0089          GOTO LOOP13$                                               2820      0278
 0090       ROOT6.                                                        2830      0279
 0090       IF COUNT-SUB GQ 4 $                                           2840      0280
 0091        BEGIN                                                        2850      0281
 0092         DIRECT                                                      2860      0282
 0093              STZ      SUFFIX                                        2870      0283
 0094         JOVIAL                                                      2880      0284
 0095         COUNT=COUNT-SUB $                                           2890      0285
 0096         GOTO LOOP12$                                                2900      0286
 0097        END                                                          2910      0287
 0098       IF COUNT GQ 5 $                                               2920      0288
 0099        GOTO LOOP13$                                                 2930      0289
 0100       ROOT8.                                                        2940      0290
 0100       IF COUNT ''STILL'' GR 12 $ ''NO STEMS > 12 CHARS. LONG''      2950      0291
 0101        TEST W $                                                     2960      0292
 0101       ''START DOUBLE LETTER ELIMINATION (VOWELS EXCLUDED)''         2970      0293
 0102       IF BYTE($COUNT-2,1$)(WUNWRD) EQ BYTE($COUNT-1,1$)(WUNWRD) $   2980      0294
 0103        BEGIN                                                        2990      0295
 0104         NXBYTE=BYTE($COUNT-1,1$)(WUNWRD)$                           3000      0296
 0105         DIRECT                                                      3010      0297
 0106              LDA      NXBYTE                                        3020      0298
 0107              SZN      ELIM+1,AL                                     3030      0299
```

Figure VI. 29. (Continued)

```
21064 01  05-14-71   17.105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0108              TNZ     RTSO                                                       3040                      0300
 0109              LDA     COUNT                                                      3050                      0301
 0110              SBA     1,DL                                                       3060                      0302
 0111              STA     COUNT                                                      3070                      0303
 0112         JOVIAL                                                                  3080                      0304
 0113        END                                                                      3090                      0305
 0114       RTSO.  ''GET FINAL STEM''                                                 3100                      0306
 0114        BYTE($0,COUNT$)(STEM)=BYTE($0,COUNT$)(WUNWRD) $                          3110                      0307
 0115        WRDENT($UNQRTS$)=WUNWRD$''KEEP A COPY OF THE WORD TOO''                  3120                      0308
 0116        STMENT($UNQRTS$)=STEM$'' A COPY OF THE STEM''                            3130                      0309
 0117        COUNTER($UNQRTS$)=COUNTER($W$) $''CHARACTER COUNT TOO''                  3140                      0310
 0118        DIRECT                                                                   3150                      0311
 0119              AOS     UNQRTS                                                     3160                      0312
 0120              STZ     STEM                                                       3170                      0313
 0121              STZ     STEM+1                                                     3180                      0314
 0122              STZ     SUFFIX                                                     3190                      0315
 0123         JOVIAL                                                                  3200                      0316
 0123                                                                                 3210                      0317
 0124       END ''"W"-LOOP .. ALL("WORDS") PROCESSED''                                3220                      0318
 0124                                                                                 3230                      0319
 0124       ''NOW SORT ALL TABLES, USING "STEMS" AS THE SORTKEY''                     3240                      0320
 0126       NENT(WORDS)=UNQRTS$ NENT(STEMS)=UNQRTS $                                                            0321
 0127       NENT(COUNTS)=UNQRTS $                                                     3251                      0322
 0128       LL=2$                                                                     3260                      0323
 0129       SRT1.        IF LL GQ UNQRTS $                                            3270                      0324
 0129                                                                                 3280                      0325
 0130                    GOTO SRT2 $                                                  3290                      0326
 0131                    LL=LL*2       $                                              3300                      0327
 0132                    GOTO SRT1 $                                                  3310                      0328
 0133       SRT2.        DD=(LL/2)=1 $                                                3320                      0329
 0136       SRT3.        II=0 $ JJ=0 $ KK=0 $                                         3330                      0330
 0137                    DIRECT                                                       3340                      0331
 0138       SRT4   LDA       II                                                       3350                      0332
 0139              ALS       1                                                        3360                      0333
 0140              EAX6      PSTEM,*AL                                                3370                      0334
 0141              LDAQ      0,6                                                      3380                      0335
 0142              STAQ      W2                                                       3390                      0336
 0143              LDA       II                                                       3400                      0337
 0144              ADA       DD                                                       3410                      0338
 0145              ALS       1                                                        3420                      0339
 0146              EAX4      PSTEM,*AL                                                3430                      0340
 0147              LDAQ      0,4                                                      3440                      0341
 0148              CMPAQ     W2                                                       3450                      0342
 0149              TNC       SRT7                                                     3460                      0343
 0150                    JOVIAL                                                       3470                      0344
 0151       SRT5. IF II+DD LS UNQRTS=1 $ ''TEST FOR SORT END''                        3480                      0345
 0152                    BEGIN                                                        3490                      0346
 0153                    II=II+1 $                                                    3500                      0347
 0154                    GOTO SRT4 $                                                  3510                      0348
 0155                    END                                                          3520                      0349
```

Figure VI.29. (Continued)



```
21084 01  05-14-71   17.105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0156                DD=(DD-1)/2 $                                                    3530        0350
  0157                IF DD LQ 0 $                                                     3540        0351
  0158                GOTO SRT13 $                                                     3550        0352
  0159                GOTO SRT3 $                                                      3560        0353
  0160              DIRECT                                                             3570        0354
  0161       FSTEM  ZERO     0,0                                                       3580        0355
  0162       SRT7   LDAQ     0,6                LOAD STEM                              3590        0356
  0163              STAQ     W2                 STORE IN W2                            3600        0357
  0164              LDAQ     0,4                                                       3610        0358
  0165              STAQ     0,6                                                       3620        0359
  0166              LDAQ     W2                 LOAD 1ST ROOT                          3630        0360
  0167              STAQ     0,4                                                       3640        0361
  0168              JOVIAL                                                             3650        0362
  0159       WRDENT($II$)==WRDENT($II+DD$) $                                           3660        0363
  0170       COUNTER($II$)==COUNTER($II+DD$) $                                         3670        0364
  0171                JJ=II-DD $                                                       3680        0365
  0172                IF JJ LS 0 $                                                     3690        0366
  0173                GOTO SRT5 $                                                      3700        0367
  0174                KK=II $                                                          3710        0368
  0175                DIRECT                                                           3720        0369
  0176       SRT8   LDA      JJ                                                        3730        0370
  0177              ALS      1                                                         3740        0371
  0178              EAX6     FSTEM,*AL                                                 3750        0372
  0179              LDAQ     0,6                                                       3760        0373
  0180              STAQ     W2                                                        3770        0374
  0181              LDA      KK                                                        3780        0375
  0182              ALS      1                                                         3790        0376
  0183              EAX4     FSTEM,*AL                                                 3800        0377
  0184              LDAQ     0,4                                                       3810        0378
  0185              CMPAQ    W2                                                        3820        0379
  0186              TZE      SRT5                                                      3830        0380
  0187              TRC      SRT5                                                      3840        0381
  0188              LDAQ     0,6                                                       3850        0382
  0189              STAQ     W2                                                        3860        0383
  0190              LDAQ     0,4                                                       3870        0384
  0191              STAQ     0,6                                                       3880        0385
  0192              LDAQ     W2                                                        3890        0386
  0193              STAQ     0,4                                                       3900        0387
  0194              JOVIAL                                                             3910        0388
  0195       WRDENT($KK$)==WRDENT($JJ$) $                                              3920        0389
  0196       COUNTER($KK$)==COUNTER($JJ$) $                                            3930        0390
  0197                JJ=JJ-DD $                                                       3940        0391
  0198                IF JJ LS 0 $                                                     3950        0392
  0199                GOTO SRT5 $                                                      3960        0393
  0200                KK=JJ+DD $                                                       3970        0394
  0201                GOTO SRT8 $                                                      3980        0395
  0202       SRT13.  ''EXIT FROM SORTING PROCEDURE COMES HERE ''                       3990        0396
  0202       RETURN$                                                                   4030        0397
  0203       END ''MORPHOLOGICAL ANALYSIS PROCEDURE ''                                 4040        0398
  0203                                                                                 4050        0399
```

Figure VI. 29. (Continued)

```
21084 01  05-14-71   17.105   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS
JOVIAL VERSION -- NUMBER 46
  0204      TERMS                                                        5280              0400
```

Figure VI.29. (Concluded)

```
1607T 01  04-29-71   18.902   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

PROCED NUMBER
  0000      START                                                  00000060   0001
  0001      PROC NUMBER(TEMP,RESET=RESETF,ACCESS,FLAG)$            00000070   0002
  0002      TABLE TEMP V 250 S 2 $                                 00000080   0003
  0002       BEGIN                                                 00000090   0004
  0003        ITEM ACCNUM I 16 U 0 0  $                            00000100   0005
  0004        ITEM TEMPID I  9 U 0 16 $                            00000110   0006
  0005        ITEM RETRNK I  7 U 0 25 $                            00000120   0007
  0006        ITEM BIBSUP I  1 U 0 32 $                            00000130   0008
  0007        ITEM LASRET I  1 U 0 33 $                            00000140   0009
  0008        ITEM CORCOE F       1 0 $                            00000150   0010
  0009       END                                                   00000160   0011
  0009      ITEM RESET I 36 S $                                    00000170   0012
  0010      ITEM ACCESS I 36 S $                                   00000180   0013
  0011      ITEM LINE H 18 $''INPUT LINE''                         00000190   0014
  0012      ITEM WDS I 36 S P 3 $                                  00000200   0015
  0013      ITEM CHS I 36 S P 18 $                                 00000210   0016
  0014      ITEM NXCHAR H 6 $''NEXT CHARACTER IN STREAM''          00000220   0017
  0015      ITEM NFAKE I 36 S $                                    00000230   0018
  0016      OVERLAY NXCHAR=NFAKE$                                  00000240   0019
  0017      ITEM DASH H 6 P 6H(00000-)$                            00000250   0020
  0018      ITEM BLNK H 6 P 6H(00000 )$                            00000260   0021
  0019      ITEM AA   H 6 P 6H(00000A)$                            00000270   0022
  0020      ITEM TT   H 6 P 6H(00000T)$                            00000280   0023
  0021      ITEM LOLIM I 36 S $''BOTTOM #''                        00000290   0024
  0022      ITEM HILIM I 36 S $''TOP #''                           00000300   0025
  0023      ITEM ERPNT H 6 $                                       00000310   0026
  0024      ITEM VALUE H 6 $                                       00000320   0027
  0025      ITEM ERPNTF I 36 S P O(000000004077)$                  00000330   0028
  0026      ITEM ERLINE H 24 $                                     00000340   0029
  0027      ITEM BAD H 12 P 12H(ERROR, RETRY)$                     00000350   0030
  0028      OVERLAY ERPNT=ERPNTF$                                  00000360   0031
  0029      ITEM LOBYTES H 6 $                                     00000370   0032
  0030      ITEM HIBYTES H 6 $                                     00000380   0033
  0031      ITEM FLAG S V(AC) V(TM) $ ''SELF-EXPLANATORY''         00000390   0034
  0032      ITEM LO I 36 S $                                       00000400   0035
  0034      ITEM DIF I 36 S $ ITEM PP I 36 S $                     00000410   0036
  0035      ITEM RESETF I 36 S $                                   00000420   0037
  0036      BEGIN ''NUMBER''                                       00000430   0038
  0036      ''CHECK FOR "GO-AHEAD"''                               00000440   0039
  0037      IF RESET LS 0 $                                        00000450   0040
  0038       RETURN$''BECAUSE SERIES IS DONE''                     00000460   0041
  0039      IF RESET GR 0 $                                        00000470   0042
  0040       GOTO GETNEXT$''BECAUSE WE'RE IN A SERIES''            00000480   0043
  0040      ''ELSE A NEW "NUMBER" OR SERIES OF NUMBERS IS NEEDED'' 00000490   0044
  0041      NUM0. ''RECEIVE TELETYPE(R) INPUT''                    00000500   0045
  0041      RECV(=LINE,WDS,CHS)$                                   00000510   0046
  0041      ''AND (RE)INITIALIZE OTHER CELLS, ETC.''               00000520   0047
  0042      LO=0$                                                  00000530   0048
  0043      HI=0$                                                  00000540   0049
```

Figure VI.30. NUMBER Listing

```
1607T 01  04-29-71   15.902   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0043       ''NOW, TRY FOR FIRST GOAL''                                            00000550        0050
 0044       FOR P=P,1,17$''FIRST GOAL SEARCH''                                     00000560        0051
 0045       BEGIN ''1ST GOAL .. ("A","T", OR "TEMP")''                             00000570        0052
 0046        BYTE($P$)(NXCHAR)=BYTE($P$)(LINE)$''FETCH A CHARACTER''               00000580        0053
 0047        PP=P$''SAVE FOR POSSIBLE OTHER GOALS OR ERROR TRACING''               00000590        0054
 0048        IF NXCHAR EQ BLANK$                                                   00000600        0055
 0049         TEST P$''IGNORE BLANKS HERE''                                        00000610        0056
 0050        IF NXCHAR EQ AA$                                                      00000620        0057
 0051         BEGIN                                                                00000630        0058
 0052          FLAG=V(AC)$''INDICATES "ACCESSION #"''                              00000640        0059
 0053          PP=P+1$''START SCAN AT NEXT BYTE FOR GOAL 2''                       00000650        0060
 0054          GOTO GOAL2$                                                         00000660        0061
 0055         END                                                                  00000670        0062
 0056        IF BYTE($P,4$)(LINE) EQ 4H(TEMP)$''USER WANTS WHOLE "TEMP" TAB''      00000680        0063
 0057         BEGIN                                                                00000690        0064
 0058          RESETF=NEXT(TEMP)$''RESETF=# ENTRIES IN TEMP TAB''                  00000700        0065
 0059           FLAG=V(TM)$''"TEMP" ENTRIES ARE NEVER DIRECTLY ACC. NO.'S ''       00000710        0066
 0060          GOTO GETNEXT$''SKIP FURTHER GOALS AND FETCH ID.''                   00000720        0067
 0061         END                                                                  00000730        0068
 0062        IF NXCHAR EQ TT $                                                     00000740        0069
 0063         BEGIN                                                                00000750        0070
 0064          FLAG=V(TM)$''INDICATES "TEMP.ID"''                                  00000760        0071
 0065          PP=P+1$''SET FOR NEXT GOAL''                                        00000770        0072
 0066          GOTO GOAL2$                                                         00000780        0073
 0067         END                                                                  00000790        0074
 0068        GOTO ERRORS''ALL OTHER CHARACTERS CONSTITUTE ERRORS''                 00000800        0075
 0069       END ''FIRST GOAL SEARCH''                                              00000810        0076
 0070       GOTO ERRORS''FALL THROUGH TO "ERROR", IF NO 1ST GOAL''                 00000820        0077
 0071       GOAL2. ''START 2ND GOAL (UP TO 5 CONSECUTIVE DIGITS)''                 00000830        0078
 0071       LOBYTES=6H(000000)$''CLEAR STORAGE AREA''                              00000840        0079
 0072       FOR P=PP,1,17$                                                         00000850        0080
 0073        BEGIN ''GOAL2 PRE-SEARCH''                                            00000860        0081
 0074         PP=P$ ''SAVE POINTER FOR ERROR TRACING AND GOAL-SEARCH''             00000870        0082
 0075         BYTE($5$)(NXCHAR)=BYTE($P$)(LINE)$''FETCH NEXT CHARACTER''           00000880        0083
 0076         IF NXCHAR EQ BLNK$                                                   00000890        0084
 0077          TEST P $''BLANKS PERMITTED HERE''                                   00000900        0085
 0078         IF NFAKE GR 10$''NON-NUMERIC CHECK''                                 00000910        0086
 0079          GOTO ERRORS''OH OH ... BAD INPUT MATERIAL''                         00000920        0087
 0080         GOTO GOAL2A$ ''CONTINUE ONWARD TO REST OF GOAL 2''                   00000930        0088
 0081        END ''PRE-SEARCH''                                                    00000940        0089
 0082       GOTO ERRORS''NO GOAL2 PRESENT .. FATAL ERROR''                         00000950        0090
 0083       GOAL2A. ''START TRUE GOAL2 SEARCH''                                    00000960        0091
 0083       FOR P=PP,1,17$                                                         00000970        0092
 0084        BEGIN ''SECOND GOAL''                                                 00000980        0093
 0085         IF LO GR 5 $                                                         00000990        0094
 0086          GOTO ERRORS''TOO MANY NUMBERS''                                     00001000        0095
 0087         BYTE($5$)(NXCHAR)=BYTE($P$)(LINE)$                                   00001010        0096
 0088         IF NFAKE GQ 10 $''POSSIBLE "GOAL2" END-- NON-NUMERIC''               00001020        0097
 0089          BEGIN                                                               00001030        0098
 0090           PP=P$''SAVE POINTER''                                              00001040        0099
```

Figure VI. 30. (Continued)

```
1607T 01  04-29-71   19.902   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0091          GOTO GOAL3$                                                        00001050    [illegible]
 0092         END                                                                 00001060    [illegible]
 0093        BYTE($LO$)(LOBYTES)=BYTE($5$)(NXCHAR)$                               00001070    [illegible]
 0094        LC=LC+1$                                                             00001080    [illegible]
 0095        PP=P$                                                                00001090    [illegible]
 0096       END                                                                   00001100    [illegible]
 0097      GOTO ERRORS$                                                           00001110    [illegible]
 0098      GOAL3. ''NOW LOOK FOR BEGINNING OF GOAL3 (DASH)''                      00001120    [illegible]
 0098      FOR P=PP,1,17$                                                         00001130    [illegible]
 0099       BEGIN                                                                 00001140    [illegible]
 0100        BYTE($5$)(NXCHAR)=BYTE($P$)(LINE)$                                   00001150    [illegible]
 0101        IF NXCHAR EQ BLNK$                                                   00001160    [illegible]
 0102         TEST P $ ''MULLIPLE BLANKS ARE "OK"''                               00001170    [illegible]
 0103        IF NXCHAR EQ DASH $''GOAL3 SATISFIED .. GOAL 4 POSSIBLE''            00001180    [illegible]
 0104         BEGIN                                                               00001190    [illegible]
 0105          PP=P+1$                                                            00001200    [illegible]
 0106          GOTO GOAL4$                                                        00001210    [illegible]
 0107         END                                                                 00001220    [illegible]
 0108        END ''GOAL3''                                                        00001230    [illegible]
 0108      ''NO GOAL4 POSSIBLE,SINCE GOAL3 UNSATISFIED''                          00001240    [illegible]
 0109      GOTO ONLYONE$''ONLY ONE NUMBER IS PRESENT''                            00001250    [illegible]
 0110      GOAL4. ''SECOND NUMBER SEARCH''                                        00001260    0121
 0110      FOR P=PP,1,17$                                                         00001270    0122
 0111      BEGIN                                                                  00001280    0123
 0112       BYTE($5$)(NXCHAR)=BYTE($P$)(LINE)$                                    00001290    0124
 0113       IF NXCHAR EQ BLNK$                                                    00001300    0125
 0114        TEST P$''MULTIPLE BLANKS ARE "OK"''                                  00001310    0126
 0115       IF NFAKE GQ 10$                                                       00001320    0127
 0119        BEGIN PP=P$ GOTO ERRORS END ''NON-NUMERIC, NON-BLANK''               00001330    0128
 0120        PP=P$                                                                00001340    0129
 0121        GOTO GOAL4A$                                                         00001350    0130
 0122       END                                                                   00001360    0131
 0123      GOTO ERRORS''"DASH" PRESENT,BUT NO NUMBERS FOLLOW IT''                 00001370    0132
 0124      GOAL4A. ''SEARCH FOR SECOND NUMBER ''                                  00001380    0133
 0124      FOR P=PP,1,17$                                                         00001390    0134
 0125       BEGIN                                                                 00001400    0135
 0126        IF HI GR 5 $ ''TOO MANY NUMBER-CHECK''                               00001410    0136
 0130         BEGIN PP=P$ GOTO ERRORS ''TOO MANY NUMBERS'' END                    00001420    0137
 0131        BYTE($5$)(NXCHAR)=BYTE($P$)(LINE)$                                   00001430    0138
 0132        IF NFAKE GQ 10$                                                      00001440    0139
 0133         GOTO CONVRT$''ALL DONE WITH "NUMBER"''                              00001450    0140
 0134        BYTE($HI$)(HIBYTES)=BYTE($5$)(NXCHAR)$                               00001460    0141
 0135        HI=HI+1$                                                             00001470    0142
 0136        PP=P$                                                                00001480    0143
 0137       END                                                                   00001490    0144
 0138      CONVRT.  ''FINAL NUMBER CONVERSIONS''                                  00001500    0145
 0138      VALUE=6H(000000)$                                                      00001510    0146
 0138      ''NOW JUSTIFY''                                                        00001520    0147
 0139      BYTE($6-HI,HI$)(VALUE)=BYTE($0,HI$)(HIBYTES)$                          00001530    0148
 0140      HILIM=BCDBIN(VALUE)$''CONVERT TO BINARY''                              00001540    0149
```

Figure VI. 30. (Continued)

```
1607T 01  04-29-71   18.902   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

 0141      VALUE=6H(000000)$                                                    00001550        0150
 0142      BYTE($6-LO,LO$)(VALUE)=BYTE($0,LO$)(LOBYTES)$''MUST JUSTIFY''        00001560        0151
 0143      LOLIM=BCDBIN(VALUE)$''CONVERT TO BINARY''                            00001570        0152
 0144      DIF=HILIM-LOLIM$                                                     00001580        0153
 0145      IF DIF LQ 0 $                                                        00001590        0154
 0146       GOTO ERRORS''RANGE ERROR DETECTED''                                 00001600        0155
 0147       RESETF=DIF$                                                         00001610        0156
 0148        GOTO GETNEXT$                                                      00001620        0157
 0149      ONLYONE. VALUE=6H(000000)$                                           00001630        0158
 0150      BYTE($6-LO,LO$)(VALUE)=BYTE($0,LO$)(LOBYTES)$''JUSTIFY''             00001640        0159
 0151      LOLIM=BCDBIN(VALUE)$''CONVERT TO BINARY''                            00001650        0160
 0152      HILIM=LOLIM$''SET UP FOR''                                           00001660        0161
 0153      RESETF=0$     ''"GETNEXT"''                                          00001670        0162
 0154      GETNEXT. ''GET NEXT MEMBER OF A SERIES''                             00001680        0163
 0154      ACCESS=HILIM-RESET$                                                  00001690        0164
 0155      IF FLAG EQ V(AC)$                                                    00001700        0165
 0156       BEGIN ''ACCESSION NUMBER''                                          00001710        0166
 0157        IF RESET LS 0 $ ''END OF OUTPUT''                                  00001720        0167
 0158         BEGIN                                                             00001730        0168
 0159          RESET=-1$ ''MAKE SURE''                                          00001740        0169
 0160          ACCESS=-1$''END OF SERIES''                                      00001750        0170
 0161         END                                                               00001760        0171
 0162        IF RESET EQ 0 $ ''LAST OF SERIES, OR "ONLY-ONE" ''                 00001770        0172
 0163        BEGIN                                                              00001780        0173
 0164         ACCESS=HILIM $                                                    00001790        0174
 0165         RESET=-1$                                                         00001800        0175
 0166          RETURN$                                                          00001810        0176
 0167         END                                                               00001820        0177
 0168        RESET=RESET-1$                                                     00001840        0178
 0169        RETURN$                                                            00001850        0179
 0170       END ''ACCESSION NUMBER''                                            00001860        0180
 0171      FLOOP. IF FLAG EQ V(TM)$                                             00001870        0181
 0172       BEGIN ''TEMPORARY I.D.''                                            00001880        0182
 0173        IF RESET LS 0 $ ''END OF OUTPUT''                                  00001890        0183
 0174         BEGIN                                                             00001900        0184
 0175          RESET=-1$ ''MAKE SURE''                                          00001910        0185
 0176          ACCESS=-1$ ''END OF SERIES''                                     00001920        0186
 0177          RETURN$                                                          00001930        0187
 0178         END                                                               00001940        0188
 0179          IF RESET EQ 0$ ''MAKE SURE''                                     00001950        0189
 0180         BEGIN ''LAST ENTRY''                                              00001960        0190
 0181          FOR T=ALL(TEMP)$                                                 00001970        0191
 0182           BEGIN ''T''                                                     00001980        0192
 0183            IF TEMPID($T$) EQ HILIM $                                      00001990        0193
 0184             BEGIN                                                         00002000        0194
 0185              ACCESS=T $                                                   00002010        0195
 0186              RESET=-1$                                                    00002020        0196
 0187              RETURN$                                                      00002030        0197
 0188             END                                                           00002040        0198
 0189            END ''T''                                                      00002050        0199
```

Figure VI.30. (Continued)

```
1607T 01  04-29-71   18.902   JOVIAL SOURCE LISTING WITH SYNTAX ERROR INDICATIONS

JOVIAL VERSION -- NUMBER 46

  0190          ACCESS=-1$                                                           00002060
  0191          RESET=-1$                                                            00002070
  0192          RETURN$                                                              00002080
  0193      END ''LAST ENTRY''                                                       00002090
  0194      FOR T=ALL(TEMP)$                                                         00002100
  0195       BEGIN ''T - AGAIN ''                                                    00002110
  0196        IF TEMPID($T$) EQ NILIM-RESET $                                        00002120
  0197         BEGIN                                                                 00002130
  0198          ACCESS=T$                                                            00002140
  0199          RESET=RESET-1$                                                       00002150
  0200          RETURN$                                                              00002160
  0201         END                                                                   00002170
  0202       END ''T - AGAIN ''                                                      00002180
  0203     RESET=RESET-1$''TRY NEXT LOWEST T.I.D ''                                  00002190
  0204     GOTO FLOOP$                                                               00002200
  0205     END ''TEMPORARY I.D. CHECK ''                                             00002210
  0206    ERROR. ''ERROR TRACING ROUTINE''                                           00002220
  0206    ERLINE=24H(                    )$                                          00002230
  0207    BYTE($PP+3,2$)(ERLINE)=BYTE($4,2$)(ERRATF)$                                00002240
  0208    SEND(ERLINE,4,24)$                                                         00002250
  0209    SEND(BAD,2,12)$                                                            00002260
  0210    GOTO NUMOS''RETRY INPUT''                                                  00002270    0221
  0211    END''NUMBER''                                                              00002280    0222
  0212    TERM$                                                                      00002290    0223
```

Figure VI. 30. (Concluded)

SECTION VII

UPDATING THE SYSTEM: RECOMMENDATIONS AND SUGGESTIONS

VII. 1 GENERAL CONSIDERATIONS

The On-line Retrieval System was not explicitly designed to permit updating of the document base. However, when the document processing method is well understood, it is clear that additional data may be added to the system without destroying the basic analysis oriented classification method for individual abstracts. This section outlines the following topics:

1) Details of the document for input to MIC. CON.
2) Limited updating procedure to follow.
3) Procedures to follow for large-scale updating or replacement of the entire data base.

VII. 2 DETAILS OF DOCUMENT FORM FOR INPUT TO MIC. CON

For a limited update to the document collection, one in which analysis is performed only on the documents to be added, only the micro-concordances of the individual documents are needed. From these micro-concordances the document concept vector generator produces a proper characterization of the new documents, providing that a sufficient number of content stems occur in the new abstract. Care must be taken to see that accession numbers produced by the micro-concordance package are in proper sequence in relation to those already in the system.

Documents to be treated in the above-mentioned fashion must be available in HISI character code and in HISI -600 system standard tape format. Each abstract forms one logical record on the tape, and may not exceed 318 machine-words in length (1908 characters). Each abstract must terminate with a machine-word containing 77_8, zero filled, in any character position.

The 77_8 (!) is used as the "escape" or end-of-document indicator, and its absence will cause a fatal error in the execution of the micro-concordance program.

VII. 3 DETAILED PROCEDURE FOR LIMITED UPDATING

The following is a list of step necessary to adding documents to the system:

1) Verify that the document is in the correct character code and is recorded in HISI-600 standard format.
2) Make sure that the "preset" item in MIC. CON which controls the first accession number generated for each run is set correctly to begin a new series.
3) Generate a micro-concordance for each new document desired.
4) Run DCVGEN against the new micro-concordances.
5) Merge the concept-document information with that obtained from the initial system loading process.
6) Run BLD. STR to create new document concept vector inverted lists.
7) Load on-line files with newer versions that include updates.

VII. 4 POSSIBLE PROBLEMS WITH LIMITED UPDATING

There are several problems which could crop up during a limited update procedure. In the first place the new document could have no occurrences of the content stems used in characterizing the collection. In this case the desired abstract's analysis could not be added to the system. Another less serious condition is the case where only a few content stems occur in the new document, and the document concept vector contains so few components that the abstract never correlates highly enough to be retrieved. In this case, the document might well be included in the source of abstract texts so that when a more extensive update is performed, its words will be included in the analysis of the whole collection.

VII. 5 LARGE-SCALE UPDATING OF THE SYSTEM

The On-line Retrieval System is not designed for large-scale updating of the document data base. The only viable procedure is to re-run the entire batch process, including the new documents in the collection. Of course, if an entirely new data base is to be installed in the On-line system the batch process must also be re-run. When materially increasing the number of documents in the system, great care must be taken to determine that no storage areas are exceeded, least important in-core information be inadvertantly destroyed.

SECTION 8

SECTION VIII

PROJECT SUMMARY

In this report the investigators have documented in detail the implementation of the R.A.D.C. Document Retrieval System, including both the Batch System and the On-Line System. Experimental observations made during the batch processing portion have been documented to aid future research in this area. The On-Line System has been designed in a modular fashion, so that future implementation of new features may be accomplished with a minimum of effort.

Recommendations for further work in this area include the following topics:

1) System "tuning": adjustment of system parameters for best and fastest retrieval
2) Additional batch experiments in the following areas:
    a) statistical filter
    b) stem clusterer
3) Addition of new functions to the On-Line System
4) Investigation of the addition of some kind of phrase processing to the system, both Batch and On-Line
5) In coordination with additional work on the stem clustering algorithm, further investigation should be made of the semantic content of the dictionary generated by clustering.

SECTION IX

REFERENCES

1. The use of the approach, pioneered by Salton and his colleagues, is discussed at length in RADC-TR-69-304.

2. Dennis, Sally F. The design and testing of a fully automatic indexing-search system for documents consisting of expository text. Information Retrieval: A Critical View. (Thompson Book Company: Washington, D.C., 1967), George Schecter, ed., pp. 67-94.

3. Hibbard, T.N. An empirical study of minimal storage sorting. C.ACM 6, 5 (May 1963), pp. 206-213.

4. Tanimoto, T.T. An elementary mathematical theory of classification and predictions, IBM Report, November 1958.

5. Maurer, W.D. An improved code for scatter storage. C.ACM 11, 1 (January 1968), pp. 35-37.

APPENDIX A

INSTALLATION OF THE DOCUMENT RETRIEVAL SYSTEM

A.1 THE BATCH SYSTEM

A.1.1 Machine Configuration Needed

In order to install either the batch system or the on-line system an HISI-600 series computer is needed. The Batch System requires the maximum core available for an HISI-635, 256K machine words. In addition, sufficient tape drives and disc space must be available to input and output the data tapes as well as the "scratch" space for the various system sorts needed. Deck set-ups are listed in Section V of this report.

A.2 THE ON-LINE SYSTEM

A.2.1 Machine Configuration Needed

As a minimum for the installation of the On-Line System the HISI-635 must be equipped for time-sharing service, that is, at least one DATANET-30 must be available. As the software used by the retrieval system is independent of device, any remote terminal supported by the installation is suitable for use with the On-Line System, including Teletypes 33 or 35 (model 37, if supported), IBM 2741 or compatible devices, DATANET-760 CRT terminal or any compatible device supported by the installation.

A.3 SYSTEM SOFTWARE

A.3.1 Operating System Requirements

The On-Line Retrieval System operates under GECOS-III, release SDL 3.0 or later. In order for the retrieval system to operate, GE-TSS must be installed along with GERTS (the Remote Terminal Supervisor).

APPENDIX B

APPENDIX B

USER'S MANUAL: R.A.D.C. ON-LINE RETRIEVAL SYSTEM

B.1 THE ON-LINE RETRIEVAL SYSTEM

This sub-section describes the broad outlines of the R.A.D.C. On-line Document Retrieval System, including the following topics:

1) Automatic document characterization
2) The interactive environment
3) Notational conventions

B.1.1 Automatic Document Characterization

The basic tenet of this system as it was developed was that the computer was to analyze a document base and obtain from it sufficient information to create a retrieval system, without manual intervention. The analysis process is described in some detail in section V of this report. In addition, the information gathered during the batch processing phase of the project was to be used for an on-line retrieval process, one in which the computer and a human would interact in real time to enable the former to retrieve relevant documents from a collection.

B.1.2 The Interactive Environment

The document retrieval system operates in an interactive environment, that is, the user and the computer program are in constant communication through a telecommunications device. Although the computer performs the actual work of retrieval, the user is in constant control of the process as he operates in conversational mode through a man-machine dialogue. For this reason the system-to-user interface is termed the Dialogue Processor.

To a large degree the Dialogue Processor is an intelligent automaton having the ability to detect errors made by the user or to hold up his progress until he makes a legal answer to a question. Much of the material of the dialogue is tutorial in nature, although, the more experienced user has the option of drastically reducing the amount of the dialogue by switching to the "terse" mode. In addition, the sophisticated user may adjust his environment to some degree through the setting of optional flags. This feature is covered in the more detailed notes below.

In the man-machine environment the user has complete control over the actions of the machine, extending or restricting his search as he sees fit, erasing his query, or looking at any document presently contained in the system, as long as he knows its accession number. Other user functions are available to enable the searcher to obtain more information about the inner working of the retrieval system, if he desires.

B. 1. 3 Notational Conventions

The following is a list of the notational conventions used in this appendix for describing the available user functions. Each paragraph which describes a user function may have one of the above indicators below, showing into which category the function fails.

- * not needed by the beginner in order to perform retrievals from the system. Called automatically.
- ** may be rendered inoperable in later versions (these are system designers' aids).

The beginner is not prevented from using the more advanced options. In fact it is exactly the intent of the designers that the opposite be true: the beginner becomes experienced only through use of the more advanced features of the system. If at any time the beginner becomes lost, he has only to return to the tutorial dialogue type and ask for HELP in order to obtain an explanation of each of the available functions.

On the other hand, some of the features of the system are not made part of the on-line messages listing for HELP, simply because they are mostly present to aid the designers in "tuning" the system. These features have been listed in this appendix for references purposes, but it is not recommended that any listed with the preface *** be used unless extreme care is employed.

B.2 WHAT THE SYSTEM WILL AND WILL NOT DO

The basic purpose of the On-line System is document retrieval, nothing else. The interactive environment allows the user to browse through the document collection, as if he were in the stacks of a library. Once have found a suitable document, he may then search out others like it. Queries to the system are made in plain English or with just isolated words, as the user desires. However, since common words are eliminated from consideration early in the analysis process, the English language equivalents of Boolean operators (and, not, or) can never have any effect on a query.

It must be remembered, however, that the user is not communicating with another human at the other end. There will be times when the program seems obstinate or obdurate: that is the nature of a fixed algorithm. The Dialogue Processor cannot second guess the user; exact replies in the correct format are essential. For this reason examples are sprinkled liberally throughout the section on user options.

Despite the inherent speed of computer systems, document retrieval is a highly complex matter. In addition, the Dialogue Processor operates in a multi-programming environment, i.e., one in which many users share the machine's resources. For the above reasons, response to a query is not immediate, and is somewhat dependant upon the length of the query (document-document search takes additional time), as well as factors related to machine utilization at the moment the query is processed. The retrieval system is fast, but it will not perform miracles.

B.3 HOW TO USE THE SYSTEM

This sub-section includes discussions of all the available user options, as well as general "tips" on how to get the maximum good out of the Dialogue Processor. It is worthwhile at this point, also, to include some conventions having to do with the usage of remote telecommunications.

1) The user may type in information to the system only after the characters 05? are sent. If the terminal has a "bell" device, it will ring when input is expected. Attempting to type in at any other time will either be ignored or may disconnect the terminal, but in no case will be recognized by the program.
2) Each line of user input is terminated by a "carriage return." The program takes care of paper advance for you.
3) Do NOT type in a "null" line, i.e., "carriage return" alone. The GECOS-III library routines used for DAC terminal access cannot handle this condition and cause an endless loop. The only possible exit from this error is the manual intervention of the HISI-635 machine operator.
4) Do NOT use the "break" key on the terminal, if there is one. "Break" will terminate the program abnormally, but does not require the intervention of the computer operator.
5) When entering query text a space character is necessary before typing the "escape" character, (exclamation point). Unless this rule is followed, the last text word will be ignored by the program.
6) Paper tape input may be prepared to simulate man-machine dialogue, if it terminates each line with the following four characters: Carriage-return, X-OFF, Rubout, Rubout.

B.3.1 How to Sign onto the System.

Details of this sub-section are necessarily vague, because actual file names, user ID's, and passwords vary from installation to installation. There are two possible conditions under which the system may be started up:

1) Initial dial-up.
2) Restart after the retrieval system has already been used once, (user is still in the time-sharing system).

The first condition is treated in the following:

1) Dial up the TSS time-sharing system number assigned.
2) Use the CARDIN system.
3) Follow one of the two available procedures:
    a) OLD the filename given by the installation; then type RUN.
    b) NEW then type RUN filename.
4) In response to the question CARD FORMAT, DISPOSITION? type NORM(;), TALK.

After the delay necessary to allocate the job for execution and any other operator action to be performed, the on-line retrieval system will sign on. The user follows directions from then on.

The second condition is treated in the following:

1) If the assigned filename for the Dialogue Processor is already in the current file, type RUN, then follow the sequence given in 4) above.
2) Else type RUN filename and follow sequence in 4) above.

In any case the On-line System will now be in control instead of the time-sharing system.

B.3.2 Description of Available Options

Each of the available user options is detailed below. Each function is listed, along with suggestions for its use, restrictions, examples of usage, and notations as to the relative level of sophistication required of its user, (T: tyro, E: experienced).

B.3.2.1 ENDF. End the current query. Clears all tables associated with query formation, query processing, and retrieval. Classification : T .

B.3.2.2 MOD. Modify the current query; add or delete words from the query; replace all of the current query. The MOD option is a method of creating a new query without destroying any of the contents of the temporary file (TEMP.) Classification : T .

B.3.2.3 DOC. Examine the bibliographies and abstracts of retrieved documents by temporary identification or any document of known accession number. For temporary file identification use the format T1 for temporary number one; A357 for accession number 357.

B.3.2.4 HELP. Obtain a list of the available options. This option furnished the user with a complete explanation of all the "public" options when in verbose dialogue, and lists the option names when in terse dialogue. At the termination of the information the user re-enters at a point where an option name is required.

*B.3.2.5 OFF. Prepare an off-line listing of the retrieved documents. This option is not implemented.

*B.3.2.6 CHG. Change the "mode" flags. Five modes are accessible by the user:

mode 1: toggle the setting for verbose-terse dialogue
mode 2: skip initial query formation from words
mode 3: print query information before retrieving
mode 4: print TEMP table information after retrieving.
mode 5: assume sophisticated user. Use of this option allows the user to respond directly to a Yes-No question with an option name instead of having to type OPTIONS before using an option name. Also, when mode switch 5 is on, explanations for all options will be given after a cry for HELP, instead of just the "tyro" options.

The following is an example of setting all the options on: 1,2,3,4,5,0 Note that the series is terminated with a "zero," and that there are no embedded blanks. Commas are used for delimiters. A response of just "zero" is possible, if the user has mistakenly entered the routine and wants to leave the modes set "off".

*B. 3. 2. 7 CON. This option allows the user to obtain the document concept vector for any document just retrieved or any document of known accession number. Form of the specification is the same as for DOC.

*B. 3. 2. 8 RET. Initiate a retrieval with the current query concept vector.

*B. 3. 2. 9 DEL. Delete from the TEMP table any document contained in it. Only documents retrieved may be treated with this function. The form of the specification is the same as for DOC.

*B. 3. 2. 10 SEE. This option permits the user to see or modify the words forming the current query.

*B. 3. 2. 11 CLR. This option clears the query concept vector to zero values for all concepts. The contents of WORDS, STEMS, and other important tables are not touched.

*B. 3. 2. 12 WRD. Obtain a list of the current query words.

*B. 3. 2. 13 DDC. Perform document-document searching. Briefly, based on the document identification typed in by the user, this function uses the document concept vector for the specified document in place of a query generated by analysis of query words. Document specification is the same as for DOC.

*B. 3. 2. 14 WGT. See or directly modify the query concept vector. In order to modify the vector, specifications must be entered by the user. The following are valid examples of input: C0003, W0. 468 or C1437,W-0. 239 or C1234, W6. 000. Notice that the fields are rigidly defined; C stands for concept number, while W stands for the weight which will replace the current weight of the indicated concept number.

**B. 3. 2. 15 ENDRUN. Completely terminate the use of the Dialogue Processor without any further delay; return to the time-sharing system.

**B.3.2.16 <u>BLD</u>. Force a re-build of the query concept vector from the current query words. Sometimes used in extensive manual query generation to assure that the retrieval will be performed against the most current query.

**B.3.2.17 <u>CHAT</u>. Return to the "CHAT" function (always called by the main Dialogue Processor elements). This function may be used to see the TEMP table after a retrieval.

**B.3.2.18 <u>SKIM</u>. Using the syntax of the concept-weight pair input used for WGT (the concept field is ignored), a lower limit of correlation value is set for the TEMP table. After a SKIM the TEMP table is compressed, and all entries lower than the skimmed value are removed from the table. The user is informed of how many items remain in TEMP after the skimming operation.

B.4 SAMPLE COMPLETE DIALOGUES WITH THE ON-LINE RETRIEVAL SYSTEM

The figures following are reproductions of actual man-machine dialogues and are presented without commentary. More detailed running commentary may be obtained in section IV.4 of this report.

```
R.A.D.C. 0N-LINE D0CUMENT RETRIEVAL SYSTEM "SIGNING 0N."
 ABSTRACTS & BIBLI0GRAPHIES ARE C0PYRIGHT 1967-1970
 AMERICAN PSYCH0L0GICAL ASS0CIATI0N. REPR0DUCTI0N PR0HIBITED.
  PLEASE AWAIT FURTHER INSTRUCTI0NS FR0M THE SYSTEM. THANKS.


   *VERSI0N--05/13/71
   IS N0RMAL SYSTEM 0PERATI0N DESIRED ?
   05?YES
   ENTER W0RDS (FREE TEXT) F0R INITIAL SEARCH QUERY, F0LL0WED
   BY "EXCLAMATI0N-P0INT"
   05?F00D DEPRIVATI0N !
   ***ANALYZING QUERY***
   ***QUERY ANALYSIS C0MPLETED***
   ***QUERY-D0CUMENT C0RRELATI0N BEGUN***
          TEMP. C0UNT
```

```
    64
AT LEAST 64 DØCUMENTS MEET THE SPECIFICATIØNS FØR THE PRESENT
QUERY
DØ YØU WANT BIBLIØGRAPHIC INFØRMATIØN FØR SØME ØF THE RETRIEVED
DØCUMENTS ?
05?YES
ACCES#    330
TEMP#       1
RANK        1
SUPPRS      0
LASRET      1
CØRREL   .820
     01A374 02Z5.6.0 05ALØRE, RICHARD K. 05BGERSHAW, N. JANE 06ZDØ
UGLASS CØLL., RUTGERS U. 07ZTHE EFFECT ØF DIFFERENTIAL FEEDING HIS
TØRIES ØN THE DISTRIBUTIØN ØF HØME CAGE ACTIVITY. 09ZPSYCHØNØMIC S
CIENCE 10Z1966, 6(5), 213-214.00 11ZJØURNAL ABSTRACT. 16A17760 19A
+ FØØD DEPRIVATIØN + FEEDING HISTØRY 16B17780 19BACTIVITY + FØØD D
EPRIVATIØN + FEEDING HISTØRY 16C10180 19C+ FØØD DEPRIVATIØN + FEED
ING HISTØRY, RAT 16D13430 19DFEEDING HABITS, FØØD DEPRIVATIØN + AC
TIVITY, RAT 16E13790 19E+ ACTIVITY + FEEDING HISTØRY, RAT 16F12940
 19FHABITS + FØØD DEPRIVATIØN + ACTIVITY, RAT 31Z41 32Z1967 33ZCØP
YRIGHT 1967 BY THE AMERICAN PSYCHØLØGICAL ASSØCIATIØN 00INC 00000

MØRE?
05?YES
ACCES#    312
TEMP#       2
RANK        2
SUPPRS      0
LASRET      1
CØRREL   .671
     01A354 02Z4.16.2 05AESSMAN, WALTER B. 05BFRISØNE, JØHN D. 06Z
QUEENS CØLL., CITY U. NEW YØRK 07ZISØLATIØN-INDUCED FACILITATIØN 0
F GASTRIC ULCERØGENESIS IN MICE. 09ZJØURNAL ØF PSYCHØSØMATIC RESEA
RCH 10Z1966, 10(2), 183-188. 100001ZW. G. SHIPMAN. 16A16070 19AULC
ERØGENESIS, ISØLATIØN + FØØD DEPRIVATIØN 16B20220 19BISØLATIØN + F
ØØD DEPRIVATIØN, MØUSE 16C13790 19CULCERØGENESIS, + ISØLATIØN, MØU
SE 16D14950 19DULCERØGENESIS, + FØØD DEPRIVATIØN, MØUSE 31Z41 32Z1
967 33ZCØPYRIGHT 1967 BY THE AMERICAN PSYCHØLØGICAL ASSØCIATIØN IN
0000C 00000
MØRE?
05?YES
ACCES#    998
TEMP#       3
RANK        3
SUPPRS      0
LASRET      1
CØRREL   .627
```

```
     01A1260 02Z5.0.0 05ALEVITT, RØBERT A. 05BWEBB, WILSE B. 06ZU.
 PITTSBURGH 07ZVARIABLES ASSØCIATED WITH A PENTØBARBITAL-INDUCED S
 LEEP RESPØNSE. 09ZPSYCHØNØMIC SCIENCE 10Z1966, 6(9), 431-4320. 11Z
JØURNAL ABSTRACT. 16A18890 19AEXPERIMENTAL VARIABLES, PENTØBARBITA
L INDUCED SLEEP, RAT 16B17750 19BEXPERIMENTAL VARIABLES + PENTØBAR
BITAL INDUCED SLEEP 16C12893 19CPENTØBARBITAL INDUCED SLEEP, + EXP
ERIMENTAL VARIABLES, RAT 31Z41 32Z1967 33ZCØPYRIGHT 1967 BY THE AM
ERICAN PSYCHØLØGICAL ASSØCIATIØN INC00000 00000
MØRE?
05?NØ
ENTER ØPTIØN NAME. (ØR "HELP" TØ SEE THE LIST ØF AVAILABLE
ØPTIØNS)
05?DØC
PRINT ØNLY RANKING AND BIBLIØGRAPHIC DATA (NØ ABSTRACTS) FØR
DØCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING
BIBLIØGRAPHIC DATA ALREADY PRINTED ?
05?NØ
SPECIFY FIRST DØCUMENT ØR DØCUMENT GRØUP TØ PRINT.
05?T1
ACCES#  330
TEMP#     1
RANK      1
SUPPRS  YES
LASRET  YES
CØRREL  .820
BIBLIØGRAPHIC DATA PRINTED  BEFØRE DURING THIS QUERY SEQUENCE
.. PRINT IT AGAIN ?
05?NØ
 PRINT THE ABSTRACT ?
05?YES
THE TEMPØRAL DISTRIBUTIØN ØF HØME CAGE ACTIVITY FØLLØWING FØØD DEP
RIVATIØN WAS INFLUENCED BY BØTH PRIØR DEPRIVATIØN EXPERIENCE AND A
N ANIMAL'S PREVIØUSLY ACQUIRED FØØD GETTING HABITS.   00000
CØNTINUE ?
05?YES
SPECIFY FIRST DØCUMENT ØR DØCUMENT GRØUP TØ PRINT.
05?T2
ACCES#  312
TEMP#     2
RANK      2
SUPPRS  YES
LASRET  YES
CØRREL  .671
BIBLIØGRAPHIC DATA PRINTED  BEFØRE DURING THIS QUERY SEQUENCE
.. PRINT IT AGAIN ?
05?NØ
PRINT THE ABSTRACT ?
05?YES
```

```
USING 2 DISCREPANT PERIØDS (8 AND 30 DAYS) WITH AGE AND DURATIØN Ø
F FØØD DEPRIVATIØN REMAINING CØNSTANT, THE DIFFERENTIAL EFFECTS ØF
 ISØLATIØN AND AGGREGATIØN UPØN THE DEVELØPMENT ØF GASTRIC PATHØLØ
GY IN MICE WERE INVESTIGATED. ''THE RESULTS SUGGEST THAT ISØLATIØN
 WHETHER BRIEF AND INTRØDUCED IN THE ADULT MØUSE, ØR PRØLØNGED AND
 INTRØDUCED JUST FØLLØWING WEANING, CØNTRIBUTES SIGNIFICANTLY TØ U
LCERØGENESIS INDUCED BY FØØD DEPRIVATIØN IN THE ADULT ANIMAL.''
00000
CØNTINUE ?
05?YES
SPECIFY FIRST DØCUMENT ØR DØCUMENT GRØUP TØ PRINT.
05?T3
ACCES#  998
TEMP#     3
RANK      3
SUPPRS  YES
LASRET  YES
CØRREL  .627
BIBLIØGRAPHIC DATA PRINTED  BEFØRE DURING THIS QUERY SEQUENCE
.. PRINT IT AGAIN ?
05?NØ
PRINT THE ABSTRACT ?
05?YES
THE EFFECT ØF A VARIETY ØF EXPERIMENTAL VARIABLES ØN THE LATENCY Ø
F A PENTØBARBITAL-INDUCED SLEEP RESPØNSE WAS STUDIED IN THE RAT. D
IURNAL CYCLE, WHITE NØISE, AND SLEEP DEPRIVATIØN WERE FØUND TØ EXE
RT SIGNIFICANT INFLUENCES WHILE BRIGHT LIGHT, SØCIAL FACILITATIØN,
 AND FØØD DEPRIVATIØN DID NØT. IN 2 STUDIES WHICH MEASURED SLEEP L
ENGTH, DIURNAL CYCLE BUT NØT FØØD DEPRIVATIØN WAS FØUND TØ EXERT A
 SIGNIFICANT EFFECT. A LØW CØRRELATIØN BETWEEN SLEEP LATENCY AND L
ENGTH SUPPØRTS THE ASSUMPTIØN THAT THESE ARE INDEPENDENT RESPØNSES
0.   00000
CØNTINUE ?
05?YES
SPECIFY FIRST DØCUMENT ØR DØCUMENT GRØUP TØ PRINT.
05?T4
ACCES#  776
TEMP#     4
RANK      4
SUPPRS  NØ
LASRET  YES
CØRREL  .601
     01A882 02Z11.14.0 05AGØRBØV, F. D. 05BKØSMØLINSKII, F. P. 05C
MYASNIKØV, V. I. 07ZNEKØTØRYE ØSØBENNØSTI VØZDEISTVIYA NA ØRGANIZM
 CHELØVEKA PØVYSHENNØI I PØNIZHENNØI AFFERENTATSII V ASPEKTE KØSMI
CHESKØI PSIKHØFIZIØLØGII. 08ZSØME EFFECTS ØF INCREASED AND DECREAS
ED SENSØRY INPUT ØN THE HUMAN ØRGANISM FRØM THE PØINT ØF VIEW ØF S
PACE PSYCHØPHYSIØLØGY. 09ZVØPRØSY PSIKHØLØGII 10Z1966, NØ. 5, 67-7
1. 11Z
PRINT THE ABSTRACT ?
05?YES
```

```
THE RESULTS OF STUDIES OF HUMAN PERFORMANCE UNDER CONDITIONS OF SE
NSORY DEPRIVATION AND SENSORY OVERLOAD ARE DISCUSSED FROM THE POIN
T OF VIEW OF THEIR RELEVANCE TO SPACE FLIGHT. THE EFFECTS OF SENSO
RY DEPRIVATION HAVE BEEN STUDIED RATHER EXTENSIVELY, WHILE RELATIV
ELY LITTLE IS KNOWN ABOUT THE EFFECTS OF SENSORY OVERLOAD. NUMEROU
S PHYSIOLOGICAL MEASURES TAKEN ON PILOTS DURING MIDAIR REFUELING O
PERATIONS SHOW THAT CONSIDERABLE STRESS IS EXPERIENCED BY THE PILO
TS. IT IS SUGGESTED THAT THE DEVELOPMENT OF OPTIMAL WORK-REST PERI
ODS AND OF SPECIAL MOTOR ROUTINES AS WELL AS THE ADMINISTRATION OF
 CERTAIN VITAMINS AND STIMULANTS IS INDICATED TO ACHIEVE A LOWERIN
G OF PSYCHOLOGICAL STRESS AND TO INCREASE PHYSIOLOGICAL TONUS.
00000
CONTINUE ?
05?NO
ENTER OPTION NAME. (OR "HELP" TO SEE THE LIST OF AVAILABLE
OPTIONS)
05?CHG
DO YOU WANT AN EXPLANATION OF THE AVAILABLE MODES ?
05?NO
"MODE" FLAGS ARE ALL OFF. IDENTIFY NUMBERS OF FLAGS TO BE SET
"ON", FOLLOWED BY A "ZERO" VALUE.  VALUES ARE COMMA DELIMITED
05?1,3,4,5,0
??
05?WRD
QUERY WORDS ARE:
FOOD
DEPRIVATION
ADD WORDS ?
05?NO
DELETE WORDS ?
05?YES
ENTER WORDS
05?FOOD !
??
05?WRD
QUERY WORDS ARE:
DEPRIVATION
ADD WORDS ?
05?YES
ENTER WORDS
05?SENSORY !
 DELETE WORDS ?
05?NO
REPLACE ALL OF PRESENT QUERY ?
05?NO
QUERY WORD ACTION ?
05?NO
DIRECT CON. VECT. ACTION ?
05?NO
RETRIEVE ?
05?NO
DOC. - DOC. RETRIEVAL ?
05?SKIM
05?C0000,W0.600
      5 ENTRIES IN "TEMP"
```

```
??
O5?CHAT
PRINT "TEMP" TABLE ?
O5?YES
 ACCESSION  TEMPORARY   CORRELATION          RANK    BIB.DATA PRINTED/
   NUMBER     IDENT.    (WHEN LAST RETRIEVED)        RET'VD LAST QUERY
     330        1           .820               1        1   /   1
     312        2           .671               2        1   /   1
     998        3           .627               3        1   /   1
     776        4           .601               4        0   /   1
    1095        5           .600               5        0   /   1
THAT'S "ALL"
PRINT BIBLIOGRAPHY ?
O5?NO
??
O5?BLD
QUERY-CONCEPT-VECTOR BUILT....
??
O5?RET
EXCLUDE PREVIOUS DOCS.?
O5?YES
SUPPRESS PREV. PRINTED ?
O5?NO
     6
TEMP. ID. STARTS WITH NUMBER --
   251
       SPACES IN "TEMP." WANT MORE ?
O5?NO
PRINT QUERY WORDS ?
O5?YES
   WORD                STEM
DEPRIVATION          DEPRIVA00000
SENSORY              SENS0000000
RETRIEVE ?
O5?YES
***ANALYZING QUERY***
***QUERY ANALYSIS COMPLETED***
***QUERY-DOCUMENT CORRELATION BEGUN***
      TEMP. COUNT
    69
NO. OF HITS >= 64
PRINT "TEMP" TABLE ?
O5?YES
```

```
  ACCESSION   TEMPORARY    CORRELATION        RANK    BIB.DATA PRINTED/
   NUMBER       IDENT.    (WHEN LAST RETRIEVED)       RET'VD LAST QUERY
     330          1            .820             1        0   /    0
     312          2            .671             2        0   /    0
     998          3            .627             3        0   /    0
     776          4            .601             4        0   /    0
    1095          5            .600             5        0   /    0
    1095          6            .554             1        0   /    1
MORE?
 05?YES
    1420          7            .550             2        0   /    1
    1414          8            .478             3        0   /    1
    1024          9            .452             4        0   /    1
    1023         10            .435             5        0   /    1
     406         11            .433             6        0   /    1
MORE?
05?DOC
TEMP. DOCS. ONLY (NO ABSTRACTS) ?
 05?NO

05?A1420
ACCES# 1420
TEMP#     7
RANK      2
SUPPRS  NO
LASRET  YES
CORREL  .550
10.0.0 05ASANCHEZ-HILDALGO, E. 06ZU. PUERTO RICO 07ZPSICOLOGIA Y C
URRICULO. 08ZPSYCHOLOGY AND CURRICULUM. 09ZREVISTA MEXICANA DE PSI
COLOGIA 10Z1966, 2(9), 785-7930. 11ZD. H. SCHUSTER. 16A12970 19ACU
RRICULUM DEVELOPMENT + PSYCHOLOGY 16B17510 19B+ EDUCATIONAL CURRIC
ULUM DEVELOPMENT 31Z41 32Z1967 33ZCOPYRIGHT 1967 BY THE AMERICAN P
SYCHOLOGICAL ASSOCIATIONO INC 00000
PRINT ABSTRACT ?
05?YES
CONTRIBUTIONS OF PSYCHOLOGY TO THE DETERMINATION OF CURRICULA ARE
DISCUSSED FROM THE VIEWPOINTS OF MENTAL DEVELOPMENT, TESTS, CULTUR
AL DEPRIVATION, FAMILY ENVIRONMENT, LANGUAGE DEVELOPMENT, AND MOTI
VATION. 5 MAJOR EDUCATIONAL POINTS ARE TAKEN FROM J. S. BRUNER (SE
E 40 7).    00000
CONTINUE ?
05?NO
??
05?ENDF
SEQUENCE KILLED
SAME MODE ?
05?NO
QUIT SYSTEM ?
05?YES
SYSTEM LOGGING "OFF" .  IT HAS BEEN A PLEASURE WORKING WITH YOU
CLOSING FILE 05
CLOSING FILE 06
```

```
ACTIVITY TERMINATED
NØRMAL TERMINATIØN

*
```

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Informatics Inc. 4720 Montgomery Lane Bethesda, Maryland 20014	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
On-Line Retrieval II

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Documentary Report

5. AUTHOR(S) *(First name, middle initial, last name)*
Peter Kurtz, Thomas C. Lowe, Thomas A. Bakert

6. REPORT DATE May 23, 1971	7a. TOTAL NO. OF PAGES 419	7b. NO. OF REFS 5
8a. CONTRACT OR GRANT NO. F30602-70-C-0199 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) TR-71-1202-1 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (INTD) Griffiss Air Force Base New York 13440

13. ABSTRACT

This report is concerned with the implementation of two interrelated computer systems: an automatic document analysis and classification package, and an on-line interactive information retrieval system which utilizes the information gathered during the automatic classification phase. Well-known techniques developed by Salton and Dennis have been adapted for the automatic classification. This work is an extension of an effort begun under contract F30602-69-C-0038.

For those interested only in performing retrievals from the on-line console, a short instructive manual is provided in the form of Appendix B of this report. Since this report is to serve as a maintenance manual as well as a description of the work performed and general findings, program listings and other detailed information have been included.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information storage and retrieval						
On-line retrieval						
Interactive retrieval						
Man-machine dialogue						
Automatic thesaurus generation						
Concept identification						
Concept vectors						
Concept clustering						
Stem analysis						
Common words						
Document clustering						
Concordances						
Project SMART						
Automatic indexing						
Correlation measures						